

INGENIERÍA DE SERVIDORES (2022-2023)

GRADO EN INGENIERÍA INFORMÁTICA

UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Diego de Luis Ballesteros

30 de noviembre de 2022

Índice

- 1. Identificar, monitorizar y reconstruir un disco RAID que está configurado en la máquina de Ubuntu**
- 2. Obligatorio: Instalación de Zabbix y utilización de la aplicación web para monitorizar concretamente ssh y http**
 - 2.1 Instalación en Ubuntu
 - 2.2 Instalación en Rocky
 - 2.3 Configuración de los ítems en la aplicación
 - 2.4 Problemas encontrados
 - 2.5 Referencias
- 3. Obligatorio: Instalación de ansible en Ubuntu para poder hacer ping a las máquinas virtuales y ejecutar comandos básicos**
 - 3.1 Instalación de ansible
 - 3.2 Ejecución de comandos básicos
 - 3.3 Ejecución de script
 - 3.4 Problemas encontrados
 - 3.5 Referencias
- 4. Extra: timer que activa un servicio**

1. Identificar, monitorizar y reconstruir un disco RAID que está configurado en la máquina de Ubuntu

Vamos a iniciar la máquina y procederemos a hacer un `cat /proc/mdstat`. Vemos que tenemos un `md0` con dos réplicas y un `md1` con otras dos réplicas funcionando correctamente y en `up (UU)`. Entonces procedemos a hacer una desconexión en caliente del segundo disco.

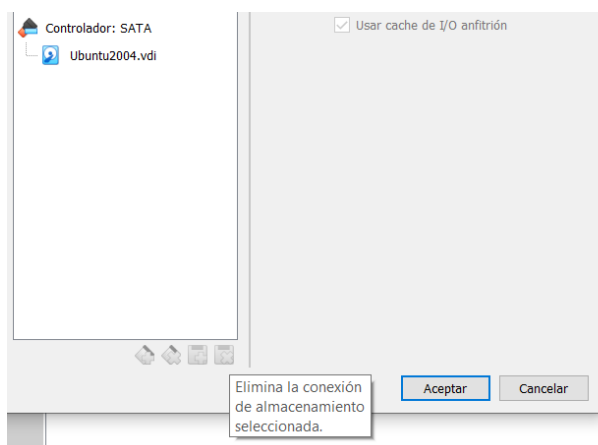


Figura 1.1

Si ahora entramos dentro de `/proc/mdstat` veremos que solo uno de los dos discos está en `up`. Por otro lado, al hacer un `lsblk` está toda la configuración correctamente pero solo de un disco. En nuestro caso hemos hecho una desconexión en caliente. Si hubiéramos hecho la desconexión y después procediéramos al arranque de la máquina habría que haber iniciado el raid desde el arranque.

Aunque sigamos teniendo el raid, nos falta uno de los discos de soporte, por tanto, vamos a crear uno nuevo de tipo `vdi` como el anterior.

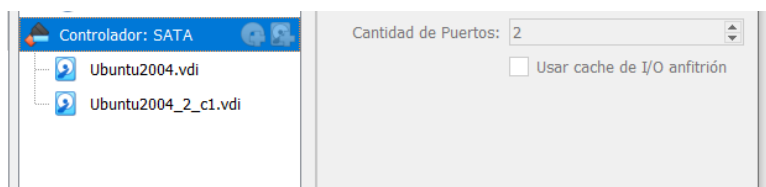


Figura 1.2

Si hacemos el `lsblk` de nuevo está el nuevo disco, pero no en el raid. Por tanto, tendremos que empezar a crear las particiones que teníamos antes para el `vdi` con `fdisk`.

Una vez creadas tendremos que introducir el grub a una maquina en `sdb` para poder tener el arranque con `sudo grub-install /dev/sbd`. Finalmente añadimos las particiones al raid con `sudo mdadm -add /dev/md0`

/dev/sdb2 e igual para el sdb3. Se podrá ver el progreso con `watch -n 1 cat /proc/mdstat` (figura 1.4).

```
diego@ubuntu:~$ sudo mdadm --add /dev/md0 /dev/sdb2
mdadm: added /dev/sdb2
diego@ubuntu:~$ practicas,ise
practicas,ise: command not found
diego@ubuntu:~$ cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[2] sda2[1]
      408576 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sda3[1]
      10063872 blocks super 1.2 [2/1] [_U]

unused devices: <none>
diego@ubuntu:~$ _
```

Figura 1.3

```
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[2] sda2[1]
      408576 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sdb3[2] sda3[1]
      10063872 blocks super 1.2 [2/1] [_U]
[====>.....] recovery = 28.4% (2862528/10063872) finish=1.2min speed=92339K/sec

unused devices: <none>
```

Figura 1.4

El `lsblk` finalmente queda así como en la figura 1.5.

```
diego1b@ubuntu:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                              7:0      0  55.6M  1 loop  /snap/core18/2632
loop1                              7:1      0  63.2M  1 loop  /snap/core20/1695
loop2                              7:2      0  55.6M  1 loop  /snap/core18/2620
loop3                              7:3      0  67.8M  1 loop  /snap/lxd/22753
loop4                              7:4      0  49.7M  1 loop  /snap/snapd/17576
loop5                              7:5      0  63.2M  1 loop  /snap/core20/1634
loop6                              7:6      0  71.3M  1 loop  /snap/lxd/16099
loop7                              7:7      0   48M   1 loop  /snap/snapd/17336
sda                                 8:0      0  10G   0 disk
├─sda1                             8:1      0    1M   0 part
├─sda2                             8:2      0  400M   0 part
├─md0                             9:0      0  399M   0 raid1 /boot
├─sda3                             8:3      0   9.6G   0 part
├─md1                             9:1      0   9.6G   0 raid1
│   └─dm_crypt-0                 253:0    0   9.6G   0 crypt
│       ├──vgraid1-swap          253:1    0    1G   0 lvm   [SWAP]
│       ├──vgraid1-home          253:2    0  500M   0 lvm   /home
│       └─vgraid1-raiz           253:3    0   8.1G   0 lvm   /
sdb                                 8:16     0   10G   0 disk
├─sdb1                             8:17     0    1M   0 part
├─sdb2                             8:18     0  400M   0 part
├─md0                             9:0      0  399M   0 raid1 /boot
├─sdb3                             8:19     0   9.6G   0 part
├─md1                             9:1      0   9.6G   0 raid1
│   └─dm_crypt-0                 253:0    0   9.6G   0 crypt
│       ├──vgraid1-swap          253:1    0    1G   0 lvm   [SWAP]
│       ├──vgraid1-home          253:2    0  500M   0 lvm   /home
│       └─vgraid1-raiz           253:3    0   8.1G   0 lvm   /
sr0                                11:0     1   61M   0 rom
```

Figura 1.5

2. Instalación de Zabbix y utilización de la aplicación web para monitorizar concretamente ssh y http

2.1 Instalación en Ubuntu

Antes de hacer nada convendría tomar una instantánea de la máquina de Ubuntu dado que vamos a modificar varias cosas. Vamos a seguir los pasos que se pueden encontrar en la página web de la primera de las referencias.

En primer lugar, hacer `sudo wget` y vamos a obtener el paquete de para instalar el repositorio del que vamos a descargar todo lo necesario para Zabbix (después de ello no olvidar hacer un `sudo apt update`).

```
diego@ubuntu:~$ sudo wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1%2Bfocal_all.deb
--2022-11-04 15:58:34-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1%2Bfocal_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
2022-11-04 15:58:35 (151 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]

diego@ubuntu:~$ sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent
```

Figura 2.1

Ahora procederemos a instalar el gestor de bases de datos del que precisamos. En nuestro caso será mariadb. Antes de instalarlo no olvidar hacer `mysql secure installation`. Comando de instalación: **`sudo apt install mariadb-server mariadb-client`**. Finalmente, hay que iniciar el servicio de mysql con **`sudo systemctl start mysql.service`**.

Para crear la base de datos fácilmente he entrado en mariadb con `sudo mysql -uroot -p` y he puesto los siguientes comandos de sql:

- `create database zabbix character set utf8 collate utf8_bin`
- `create user zabbix@localhost identified by practicas,ise`
- `grant all privileges on zabbix.* to zabbix@localhost`

```
diego@ubuntu:~$ sudo mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> _

MariaDB [(none)]> create database zabbix character set utf8 collate utf8_bin
-> ;
Query OK, 1 row affected (0.129 sec)

MariaDB [(none)]> create user zabbix@localhost identified by practicas,ise
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'practicas,ise' at line 1
MariaDB [(none)]> create user zabbix@localhost identified by 'practicas,ise'
-> ;
Query OK, 0 rows affected (0.148 sec)

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.002 sec)
```

Figura 2.2

Ahora habría que instalar apache2 y ssh, apache2 ya viene instalado y ssh ya estaba instalado de la práctica anterior.

Ahora tenemos que importar la base de datos necesaria para zabbix. Para eso vamos a tocar algunos datos de la configuración con el siguiente comando de la figura 2.3.

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

Figura 2.3

Ahora tenemos que editar unos cuantos archivos de configuración para que zabbix funcione correctamente. En concreto /etc/zabbix/zabbix_server.conf (1), /etc/zabbix/zabbix_agentd.conf (2), y /etc/zabbix/apache.conf(3). En el primero tendremos que modificar la contraseña para zabbix (Figura 2.3). En el segundo debemos poner la ip de nuestra máquina y el nombre del host para la monitorización (Figura 2.4). Finalmente, en el tercero debemos añadir la zona horaria para que la aplicación funcione correctamente (Figura 2.5).

```
DBUser=zabbix

### Option: DBPassword
# Database password.
# Comment this line if no password is used.
#
# Mandatory: no
# Default:
# DBPassword=
DBPassword=practicas,ise
```

Figura 2.3

```
### Option: Server
# List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix s
# 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
# 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
```

```
ServerActive=192.168.56.105

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=
Hostname=Ubuntu Server
```

Figura 2.4

```
<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/Riga
</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/Riga
```

Figura 2.5

También deberíamos añadir el puerto 10050, que es el que usa la aplicación de zabbix por defecto con `sudo ufw allow 10050`. Por último iniciamos los servicios con **`sudo systemctl enable --now zabbix-server zabbix-agent apache2`**.

2.2 instalación en rocky

Para la instalación del agente de zabbix en este caso va a ser más sencillo que en Ubuntu. Lo primero de todo vamos a deshabilitar los paquetes de Zabbix proporcionados por EPEL. Voy a editar el archivo `/etc/yum.repos.d/epel.repo` y agregaré concretamente en la parte de `[epel]` **`"excludepkgs=zabbix*"`** (figura 2.6)

```
[epel]
name=Extra Packages for Enterprise Linux $releasever - $basearch
# It is much more secure to use the metalink, but if you wish to use a local mirror
# place its address here.
#baseurl=https://download.example/pub/epel/$releasever/Everything/$basearch/
metalink=https://mirrors.fedoraproject.org/metalink?repo=epel-$releasever&arch=$basearch&infra=$infra&content=$contentdir
enabled=1
gpgcheck=1
countme=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-$releasever
excludepkgs=zabbix*
```

Figura 2.6

Después instalaremos los nuevos repositorios de zabbix (Figura 2.7). Una vez instalados, habrá que hacer un `dnf clean` para eliminar la caché de los paquetes anteriores. Proseguimos con la instalación del agente de zabbix (**`sudo dnf install zabbix-agent`**) y la iniciación del servicio con `sudo systemctl enable --now zabbix-agent`.

```
[diegolb@localhost ~]$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
warning: /var/tmp/rpm-tmp.D7kF8j: Header V4 RSA/SHA512 Signature, key ID 08efa7dd: NOKEY
Verifying...
Preparing...
Updating / installing...
 1:zabbix-release-5.0-3.el9
```

Figura 2.7

Para finalizar, habrá que añadir la ip de nuestro host y añadir el nombre del host en `/etc/zabbix/zabbix_agentd.conf` (figura 2.8). Además, será necesario también añadir el puerto 10050 al firewall de rocky (figura 2.9).

```
### 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=192.168.56.110
```

```

ServerActive=192.168.56.110

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match h
# Value is acquired from HostnameItem if unde
###
# Mandatory: no
# Default:
# Hostname=

Hostname=Host Rocky

```

Figura 2.8

```

[dieloglb@localhost ~]$ sudo firewall-cmd --add-port=10050/tcp --permanent
[sudo] password for dieloglb:
success
[dieloglb@localhost ~]$ sudo firewall-cmd --reload
success

```

Figura 2.9

2.3 Configuración de los ítems en la aplicación

Antes de nada, será necesario acceder a la aplicación usando cualquier navegador. Dentro del navegador ponemos el siguiente enlace para acceder a la aplicación de nuestra máquina virtual: <http://192.168.56.105/zabbix/setup.php>. Nos llevará a una serie de páginas. En la tercera deberemos poner la contraseña que hayamos introducido en /etc/zabbix/zabbix_server.conf (figura 2.3). En la cuarta habrá que poner Ubuntu Server y en la última el nombre de usuario será **Admin** y la contraseña será **zabbix**.

The image displays three sequential screenshots of the Zabbix 5.0 installation web interface. Each screenshot features a sidebar on the left with navigation links: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details, Pre-installation summary, and Install.

Top Left Screenshot: Shows the 'Welcome to Zabbix 5.0' page with the 'Install' link selected in the sidebar.

Top Right Screenshot: Shows the 'Check of pre-requisites' page. It contains a table with the following data:

	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/Riga		OK
PHP databases support	MySQL		OK
PHP bcmath	on		OK
PHP mbstring	on		OK

Bottom Left Screenshot: Shows the 'Configure DB connection' page. It includes the following fields:

- Database type:
- Database host:
- Database port: (0 - use default port)
- Database name:
- User:
- Password:

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

Bottom Right Screenshot: Shows the 'Zabbix server details' page. It includes the following fields:

- Host:
- Port:
- Name:

ZABBIX

Pre-installation summary

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

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

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: Ubuntu Server

Back

Next step

ZABBIX

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 "conf/zabbix.conf.php" created.

Back

Finish

ZABBIX

Username

admin

Password

☒ Remember me for 30 days

Sign in

Figura 2.10

Ahora vamos a añadir los hosts para la monitorización de http y ssh. Para crear un host nos vamos a la parte de configuración y luego a hosts. (se puede comprobar que soy yo en la figura 2.12)

ZABBIX

Ubuntu Server

Monitoring

Inventory

Reports

Configuration

Host groups

Templates

Hosts

Maintenance

Actions

Event correlation

Discovery

Services

Administration

Support

Share

Hosts

All hosts / Ubuntu Server

Enabled

Linux servers

Applications 14

Items 112

Triggers 55

Graphs 21

Discovery rules 4

Web scenarios

Host

Templates

IPMI

Tags

Macros

Inventory

Encryption

* Host name

Ubuntu Server

Visible name

* Groups

Linux servers

Select

Type

IP address

DNS name

Connect to

Port

Default

Agent

127.0.0.1

IP

DNS

10050

Remove

Agent

192.168.56.105

IP

DNS

10050

Remove

Add

Description

Monitored by proxy

(no proxy)

Enabled

Update

Clone

Full clone

Delete

Cancel

Figura 2.11



Figura 2.12

Como se puede ver (figura 2.11) tenemos que añadir el mismo nombre que hemos puesto en /etc/zabbix/zabbix_agentd.conf (figura 2.4), en nuestro caso, **Ubuntu Server**. Además hemos de añadir como punto importantela ip de nuestra máquina (también visible en el fichero mencionado). Después le damos a templates y añadimos dos ítems, el de http y el de ssh (yo los he renombrado por comodidad. El de ssh es el único que hay y el de http es Template App HTTP Service).

Linked templates

Name	Action
HTTP	Unlink Unlink and clear
SSH	Unlink Unlink and clear

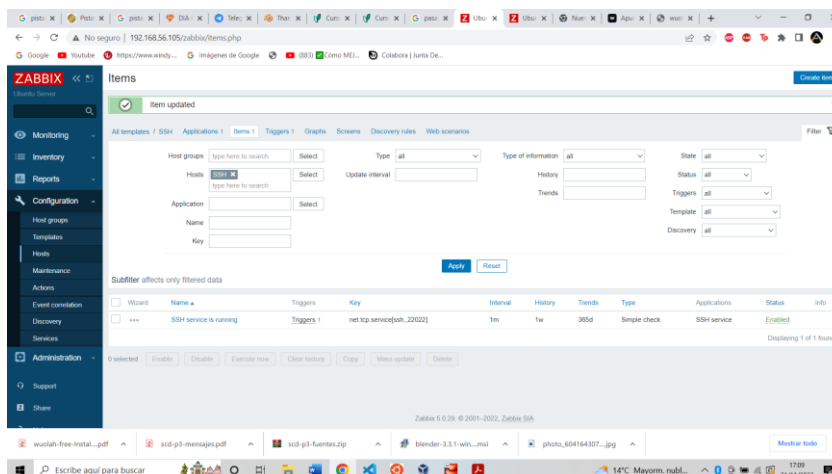
Link new templates

type here to search Select

Update Clone Full clone Delete Cancel

Figura 2.13

Ahora tenemos un problema y es que nosotros en nuestras máquinas (desde la práctica 2) tenemos configurado que el puerto por defecto de ssh sea el 22022. Por tanto, tenemos que añadirlo. Nos vamos al template de ssh y entramos en el apartado de ítems. Una vez ahí veremos que solo hay un ítem, le damos click y lo configuramos como se puede ver en la segunda página de las referencias añadiendo el puerto 22022 (el 22 estará por defecto)



* Name

Type

* Key

User name

Password

Type of information

Units

* Update interval

Figura 2.14

Por otro lado, para añadir un host para Rocky, tendremos que seguir los mismos pasos, solo que en este caso no hará falta solucionar el problema del ssh porque ya lo hemos hecho. Por último, una vez configurado, nos saldrá como en la figura 2.15 en el apartado de monitoring y hosts. Si nos vamos a la parte de graphics, en cada host, veremos que están todas las gráficas de monitorización.

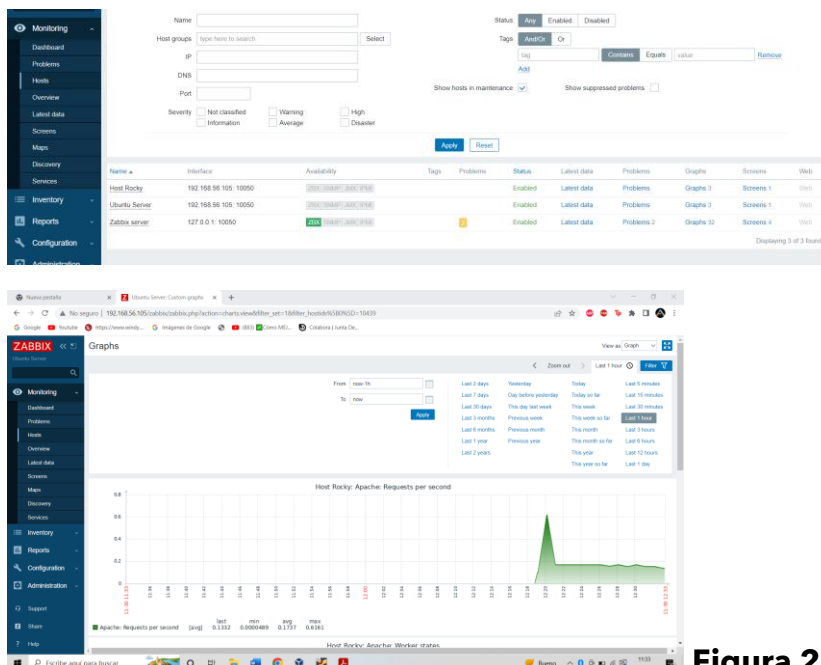


Figura 2.15

2.4 Problemas encontrados

1. Había que instalar mariaDB en Ubuntu antes de hacer nada, yo no lo hice y por tanto no podía continuar. También me faltó hacer el mysql secure installation.

2. Al intentar importar la base de datos usando el comando `zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix` no me dejó hacerlo bien (no sé por qué) así que tuve que desinstalar zabbix-server-mysql y volverlo a instalar. Sin embargo, rápidamente me di cuenta de que ese no era el problema y vi en `/var/log/zabbix/zabbix_server.log` no me detectaba la base de datos y por tanto no se podía iniciar zabbix-server. Entonces, reinicié la máquina, paré el servicio, eliminé la base de datos y sus

usuarios. Posteriormente, la volví a crear y la importé a zabbix. Inicié los servicios y ya funcionaba correctamente.

2.5 Referencias

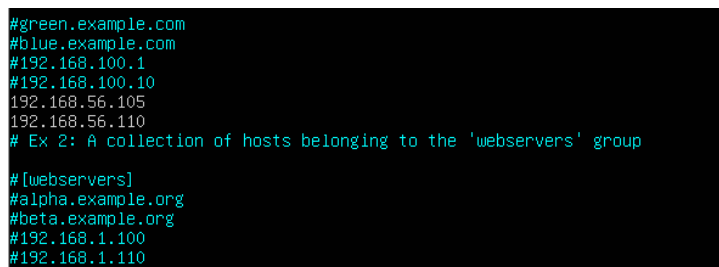
- 1- https://www.zabbix.com/download?zabbix=5.0&os_distribution=ubuntu&os_version=20.04&components=server_frontend_agent&db=mysql&ws=apache
- 2- https://www.zabbix.com/documentation/current/en/manual/config/items/itemtypes/zabbix_agent

3. Instalación de ansible en Ubuntu para poder hacer ping a las máquinas virtuales y ejecutar comandos básicos

3.1 Instalación de ansible

Ansible es una interfaz que permite programar scripts y visualizar su ejecución de una manera cómoda y visual. La instalación será más sencilla que la de zabbix, simplemente habrá que hacer un **sudo apt install ansible**.

En primer lugar, habrá que añadir los hosts que usará ansible. Para ello entraremos en /etc/ansible/hosts (figura 3.1). Se pueden poner en cualquier lugar del fichero.



```
#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
```

Figura 3.1

Como anotación, al hacer un `sudo systemctl status ansible` no va a hacer nada porque este no es un servicio. Por otro lado, al ejecutar el comando **ansible all -m ping -u diegolb** veo que no estoy conectado al puerto (figura 3.2). Para solucionar esto, lo único que habrá que hacer será editar el fichero /etc/ansible/ansible.cfg para cambiar el puerto por defecto (que será el 22) y poner el puerto 22022, ya que, si no, no nos dejará conectarnos (figura 3.3).

```
diego@ubuntu:~$ ansible all -m ping -u diegolb
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: Conn
ection 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: Conn
ection refused",
  "unreachable": true
}
```

Figura 3.2

```
[defaults]
# some basic default values...

#inventory      = /etc/ansible/hosts
#library         = /usr/share/my_modules/
#module_utils    = /usr/share/my_module_utils/
#remote_tmp      = ~/.ansible/tmp
#local_tmp       = ~/.ansible/tmp
#plugin_filters_cfg = /etc/ansible/plugin_filters.yml
#forks           = 5
#poll_interval   = 15
#sudo_user       = root
#ask_sudo_pass   = True
#ask_pass        = True
#transport       = smart
remote_port      = 22022
#module_lang     = C
#module_set_locale = False
```

Figura 3.3

Antes de volver a hacer **ansible -all -m ping -u diegolb** (Figura 3.4), deberíamos haber compartido las claves públicas de ssh también con nuestra máquina, ya que ansible se va a conectar mediante ssh a la máquina donde está instalada y a la máquina remota para este ejercicio. Veremos en el apartado de problemas que esto me ha dado ciertos problemas (también he tenido que poner el mismo nombre de usuario en las dos máquinas para poder ejecutar el comando en las dos máquinas a la vez).

```
diegolb@ubuntu:~/root$ ansible all -m ping -u diegolb
[WARNING]: Platform linux on host 192.168.56.110 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
192.168.56.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
```

Figura 3.4

3.2 Ejecución de comandos básicos

Vamos a ejecutar algún comando básico con ansible para las dos máquinas. En nuestro caso haremos un ls (figura 3.5) con **ansible all -a "ls /home" -u diegolb**. Como anotación, "all" nos indica que el comando se ejecutará en todas las máquinas. También se pueden crear grupos (por ejemplo, webserver) a los que mandárselo.

```
diegolb@ubuntu:~$ ansible all -a "ls /home" -u diegolb
192.168.56.105 | CHANGED | rc=0 >>
diego
lost+found
[WARNING]: Platform linux on host 192.168.56.110 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
192.168.56.110 | CHANGED | rc=0 >>
diego
diegolb
```

Figura 3.5

3.3 Ejecución de script

Ahora vamos a usar el script que hay en swad (mon-raid.py) para ejecutarlo en las dos máquinas. En mi caso yo lo he ubicado dentro de la carpeta de usuario en las dos máquinas.

```
#!/usr/bin/env python3

import re

error_pattern = re.compile(r'\[[0-9]{2},\}')

with open('/proc/mdstat') as f:
    for line in f:
        match = re.search(error_pattern, line)
        if not match or "-" not in match.string:
            continue
        print("--ERROR en RAID--")
        break
    else:
        print("--All RAID's OK--")

print("--OK Script--")
```

```
diegolb@ubuntu:~$ ansible all -a "python3 /home/diegolb/mon-raid.py" -u diegolb
[WARNING]: Platform linux on host 192.168.56.110 is using the discovered Python interpreter at
/usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more
information.
192.168.56.110 | CHANGED | rc=0 >>
--All RAID's OK--
--OK Script--
192.168.56.105 | CHANGED | rc=0 >>
--All RAID's OK--
--OK Script--
```

Figura 3.6

3.4 Problemas encontrados

1. el primer problema es que si quiero conectarme ahora mediante ansible vía ssh no puedo porque no he compartido la clave pública con la propia máquina donde está instalado ansible, por tanto, para ello, tengo que comentar el password authentication (figura 3.7) dentro del fichero **/etc/ssh/sshd_config** en las dos máquinas y volver a iniciar ssh. Ahora compartimos las claves publicas entre las máquinas y a las mismas máquinas (por si luego queremos utilizar ansible en rocky también), descomentamos los password authentication y reiniciamos ssh (práctica 2).

```
# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication no
#PermitEmptyPasswords no

"/etc/ssh/sshd_config" 130L, 3654B written
(diegolb@localhost ~)$ sudo systemctl restart sshd_
```

Figura 3.7

2. he tenido que cambiar el nombre de usuario de Ubuntu para que estén los dos nombres de usuario iguales y me funcione bien **ansible all -m ping -u diegolb** por ejemplo.

3.5 Referencias

- 1- https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html
- 2- https://docs.ansible.com/ansible/latest/getting_started/index.html

4. Timer que activa un servicio

Vamos a usar para ello el archivo de Python que hemos creado en el apartado 3. Ahora dentro de **/etc/systemd/system** creamos los archivos de `mon_raid.service` (figura 4.1) y `mon_raid.timer` (figura 4.2) que se pueden encontrar en swad también.

```
[Unit]
Description=Monitor RAID status

[Service]
Type=simple
ExecStart=/home/diegolb/mon-raid.py
```

Figura 4.1

```
[Unit]
Description=Monitor RAID status

[Timer]
OnCalendar=minutely

[Install]
WantedBy=timers.target_
```

Figura 4.2

Antes de nada, si no le damos permisos de ejecución al archivo de Python no va a iniciarse bien el `mon_raid.service` (figura 4.3), por tanto hacemos un `chmod +x` y listo.

```

Nov 24 19:58:11 ubuntu systemd[6827]: mon_raid.service: Failed to execute command: Permission denied
Nov 24 19:58:11 ubuntu systemd[6827]: mon_raid.service: Failed at step EXEC spawning /home/diegolb/
-- Subject: Process /home/diegolb/mon-raid.py could not be executed
-- Defined-By: systemd
-- Support: http://www.ubuntu.com/support
--
-- The process /home/diegolb/mon-raid.py could not be executed and failed.
--
-- The error number returned by this process is ERRNO.
Nov 24 19:58:11 ubuntu systemd[1]: mon_raid.service: Main process exited, code=exited, status=203/E
-- Subject: Unit process exited
-- Defined-By: systemd
-- Support: http://www.ubuntu.com/support
--
-- An ExecStart= process belonging to unit mon_raid.service has exited.
--
-- The process' exit code is 'exited' and its exit status is 203.
Nov 24 19:58:11 ubuntu systemd[1]: mon_raid.service: Failed with result 'exit-code'.
-- Subject: Unit failed
-- Defined-By: systemd
-- Support: http://www.ubuntu.com/support

```

Figura 4.3

Ahora sí, iniciamos el servicio con **systemctl start mon-raid.timer** y ya no dará error. Podemos ver la salida el timer las últimas 24 horas con **journalctl -u mon_raid --since="yesterday"** (figura 4.5).

```

diegolb@ubuntu:~/home$ cd $HOME
diegolb@ubuntu:~$ systemctl start mon_raid.timer
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'mon_raid.timer'.
Authenticating as: DiegoLB (diegolb)
Password:
==== AUTHENTICATION COMPLETE ====

```

Figura 4.4

```

-- Logs begin at Thu 2022-10-06 08:06:07 UTC, end at Fri 2022-11-25 11:58:37 UTC. --
Nov 24 19:58:11 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 19:58:11 ubuntu systemd[1]: mon_raid.service: Main process exited, code=exited, status=203/E
Nov 24 19:58:11 ubuntu systemd[1]: mon_raid.service: Failed with result 'exit-code'.
Nov 24 19:59:03 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 19:59:03 ubuntu systemd[1]: mon_raid.service: Main process exited, code=exited, status=203/E
Nov 24 19:59:03 ubuntu systemd[1]: mon_raid.service: Failed with result 'exit-code'.
Nov 24 20:00:14 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:00:14 ubuntu systemd[1]: mon_raid.service: Main process exited, code=exited, status=203/E
Nov 24 20:00:14 ubuntu systemd[1]: mon_raid.service: Failed with result 'exit-code'.
Nov 24 20:01:03 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:01:03 ubuntu systemd[6880]: mon_raid.service: Failed to execute command: Permission denied
Nov 24 20:01:03 ubuntu systemd[6880]: mon_raid.service: Failed at step EXEC spawning /home/diegolb/
Nov 24 20:01:03 ubuntu systemd[1]: mon_raid.service: Main process exited, code=exited, status=203/E
Nov 24 20:01:03 ubuntu systemd[1]: mon_raid.service: Failed with result 'exit-code'.
Nov 24 20:02:01 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:02:01 ubuntu mon-raid.py[6908]: --All RAID's OK--
Nov 24 20:02:01 ubuntu mon-raid.py[6908]: --OK Script--
Nov 24 20:02:01 ubuntu systemd[1]: mon_raid.service: Succeeded.
Nov 24 20:03:11 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:03:11 ubuntu mon-raid.py[6929]: --All RAID's OK--
Nov 24 20:03:11 ubuntu mon-raid.py[6929]: --OK Script--
Nov 24 20:03:11 ubuntu systemd[1]: mon_raid.service: Succeeded.
Nov 24 20:04:03 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:04:04 ubuntu mon-raid.py[6944]: --All RAID's OK--
Nov 24 20:04:04 ubuntu mon-raid.py[6944]: --OK Script--
Nov 24 20:04:04 ubuntu systemd[1]: mon_raid.service: Succeeded.
Nov 24 20:05:14 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:05:14 ubuntu mon-raid.py[6961]: --All RAID's OK--
Nov 24 20:05:14 ubuntu mon-raid.py[6961]: --OK Script--
Nov 24 20:05:14 ubuntu systemd[1]: mon_raid.service: Succeeded.
Nov 24 20:06:03 ubuntu systemd[1]: Started Monitor RAID status.
Nov 24 20:06:04 ubuntu mon-raid.py[6974]: --All RAID's OK--
Nov 24 20:06:04 ubuntu mon-raid.py[6974]: --OK Script--
Nov 24 20:06:04 ubuntu systemd[1]: mon_raid.service: Succeeded.
Nov 24 20:07:14 ubuntu systemd[1]: Started Monitor RAID status.

```

Figura 4.5