

ISE 22/23 - Práctica 3

P3L1 - Ejercicio recuperación RAID

Desde P1L3 (CentOs) con fallo software

1. Comprobar RAID

```
sudo mdadm --query /dev/md0 #o md127, si el SO lo reasigna
sudo mdadm --detail /dev/md0 #Para mas detalle
```

```
lgeraylp@localhost ~1$ sudo mdadm --query /dev/md127
/dev/md127: 2045.00MiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
lgeraylp@localhost ~1$
```

```
lgeraylp@localhost ~1$ sudo mdadm --detail /dev/md127
[sudo] password for lgeraylp:
/dev/md127:
    Version : 1.2
    Creation Time : Sun Oct  9 21:39:48 2022
    Raid Level : raid1
    Array Size : 2094080 (2045.00 MiB 2144.34 MB)
    Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Sat Nov 12 21:20:27 2022
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0

    Consistency Policy : resync

    Name : localhost.localdomain:0 (local to host localhost.localdomain)
    UUID : 1c321c7a:ad041917:5ee67d75:93829ce9
    Events : 35

    Number Major Minor RaidDevice State
    0      8      17        0      active sync  /dev/sdb1
    1      8      33        1      active sync  /dev/sdc1
```

2. Para ver el estado del multidevice (mdstatus)

```
cat /proc/mdstat
```

```
Personalities : [raid1]
md127 : active raid1 sdb1[0] sdc1[1]
      2094080 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Lo importante son las U: representan los discos activos (up). Un _ representaría un disco caído.

3. Forzar un fallo software

```
sudo mdadm --manage --set-faulty /dev/md0 /dev/sdb1 #Tiramos el disco 1 (su
unica partición)
```

```
[yeraylp@localhost ~]$ sudo mdadm --manage --set-faulty /dev/md127 /dev/sdb1
[sudo] password for yeraylp:
[ 911.467056] md/raid1:md127: Disk failure on sdb1, disabling device.
[ 911.467056] md/raid1:md127: Operation continuing on 1 devices.
mdadm: set /dev/sdb1 faulty in /dev/md127
```

Podemos comprobarlo con detail o mdstat (ni query ni lsblk nos darán la info necesaria)

```
sudo mdadm --detail /dev/md127 #OJO --query no muestra nada
cat /proc/mdstat
```

```
[yeraylp@localhost ~]$ sudo mdadm --detail /dev/md127
/dev/md127:
    Version : 1.2
  Creation Time : Sun Oct  9 21:39:48 2022
    Raid Level : raid1
    Array Size : 2094080 (2045.00 MiB 2144.34 MB)
  Used Dev Size : 2094080 (2045.00 MiB 2144.34 MB)
    Raid Devices : 2
    Total Devices : 2
 Persistence : Superblock is persistent

    Update Time : Sat Nov 12 21:35:34 2022
      State : clean, degraded
    Active Devices : 1
    Working Devices : 1
    Failed Devices : 1
     Spare Devices : 0

Consistency Policy : resync

    Name : localhost.localdomain:0 (local to host localhost.localdomain)
    UUID : 1c321c7a:ad041917:5cc67d75:93829cc9
    Events : 49

   Number   Major   Minor   RaidDevice State
    ---   -
    -         0         0         0   removed
    1         8        33         1   active sync    /dev/sdc1
    0         8        17         -   faulty    /dev/sdb1
```

```
[yeraylp@localhost ~]$ cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sdb1[0](F) sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]

unused devices: <none>
```

La barra baja (_) nos muestra el disco caído

4. Para recuperarnos del fallo, hay que sacar el disco caído del RAID y luego volverlo a meter

```
sudo mdadm -r /dev/md0 /dev/sdb1 #Quitamos
cat /proc/mdstat #Comprobamos

sudo mdadm -a /dev/md0 /dev/sdb1 #Añadimos
cat /proc/mdstat #Monitorizamos la recuperacion
```

```
lyeraylp@localhost ~]# sudo mdadm -r /dev/md127 /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md127
```

hot removed nos indica que se retira en caliente (con el servidor activo)

```
lyeraylp@localhost ~]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]

unused devices: <none>
```

No aparece sdb1

```
lyeraylp@localhost ~]# sudo mdadm -a /dev/md127 /dev/sdb1
mdadm: added /dev/sdb1
lyeraylp@localhost ~]# [ 1623.570868] md: recovery of RAID array md127
sudo mdadm -a /dev/md127 cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sdb1[2] sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]
      [==>.....] recovery = 19.1% (401288/2094080) finish=0.8min speed=401200K/sec

unused devices: <none>
lyeraylp@localhost ~]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sdb1[2] sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]
      [=====>.....] recovery = 32.6% (683968/2094080) finish=0.1min speed=227969K/sec

unused devices: <none>
lyeraylp@localhost ~]# cat /proc/mdstat
Personalities : [raid1]
md127 : active raid1 sdb1[2] sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]
      [=====] recovery = 66.8% (1400832/2094080) finish=0.0min speed=233472K/sec

unused devices: <none>
lyeraylp@localhost ~]# cat /proc/mdstat [ 1634.803359] md: md127: recovery done.

Personalities : [raid1]
md127 : active raid1 sdb1[2] sdc1[1]
      2094080 blocks super 1.2 [2/2] [UU]

unused devices: <none>
lyeraylp@localhost ~]# _
```

5. Podemos monitorizar la recuperación del disco con watch

```
watch -n 1 /proc/mdstat #Se sale con Ctrl+C
```

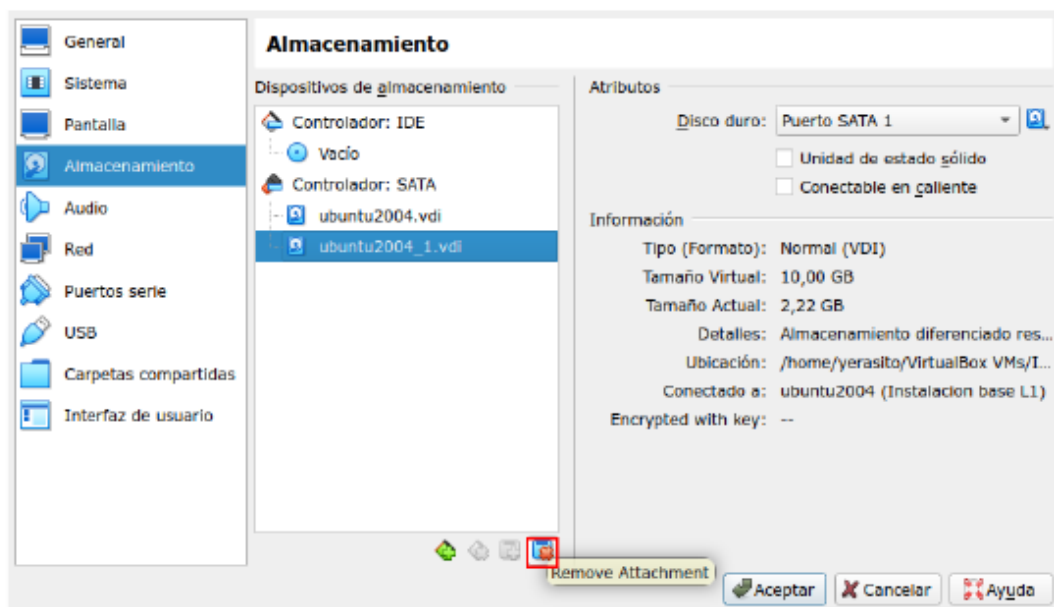
```
Every 1.0s: cat /proc/mdstat                                localhost.localdomain: Sat Nov 12 21:49:43 2
Personalities : [raid1]
md127 : active raid1 sdb1[2] sdc1[1]
      2094080 blocks super 1.2 [2/1] [_U]
      [=====] recovery = 56.2% (1178496/2094080) finish=0.0min speed=235699K/sec

unused devices: <none>
```

Podríamos monitorizarlo con un demonio, pero se recomienda el uso de herramientas específicas de monitorización como zabbix o el resto del guión

Desde P1L1 (UbuntuServer) con fallo hardware

1. Forzamos el fallo hardware desde VirtualBox



```
Cannot process volume group vg0
cryptsetup: Waiting for encrypted source device
UUID=41d79621-4e91-4f79-a323-b789f2aac693...
```

Intentará cargar el grupo de volúmenes, pero no tiene el disco. Al rato de intentar iniciar, saltará el error:

```
ALERT! encrypted source device UUID=41d79621-4e91-4f79-a323-b789f2aac693 does not exist, can't unlock dm_crypt-0.
Check cryptopts=source= bootarg: cat /proc/cmdline
or missing modules, devices: cat /proc/modules; ls /dev
Dropping to a shell.

BusyBox v1.30.1 (Ubuntu 1:1.30.1-4ubuntu6.1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)
```

Nos deja en initramfs. Con doble tab podemos ver los comandos que ofrece.

2. Podemos ver los mensajes del kernel con dmesg
3. Mirar la consola lvm - no tiene nada: discos, particiones...

```
(initramfs) lvm
lvm> lvmdiskscan
  0 disks
  0 partitions
  0 LVM physical volume whole disks
  0 LVM physical volumes
lvm> lvdisplay
lvm> vgdisplay
lvm>
(initramfs)
```

4. Si miramos mdstat vemos que si ha usado los discos, ha arrancado porque ha encontrado boot, pero están inactivos:

```
(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : inactive sda3[1](S)
      10063872 blocks super 1.2
md0 : inactive sda2[1](S)
      408576 blocks super 1.2

unused devices: <none>
(initramfs)
```

5. Forzamos la activación de md0 y md1

```
mdadm -R /dev/md0
cat /proc/mdstat
mdadm -R /dev/md1
cat /proc/mdstat
```

Forzamos el arranque desde initramfs con CTRL+D

```
(initramfs) mdadm -R /dev/md0
[ 249.712529] md/raid1:md0: active with 1 out of 2 mirrors
[ 249.713177] md0: detected capacity change from 0 to 418381824
mdadm: started array /dev/md0
(initramfs) mdadm -R /dev/md1
[ 252.307650] md/raid1:md1: active with 1 out of 2 mirrors
[ 252.308263] md1: detected capacity change from 0 to 10905404928
mdadm: started array /dev/md1
(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : active (auto-read-only) raid1 sda3[1]
      10063872 blocks super 1.2 [2/1] [_U]

md0 : active (auto-read-only) raid1 sda2[1]
      408576 blocks super 1.2 [2/1] [_U]

unused devices: (none)
(initramfs) done.
Begin: Running /scripts/local-premount ... [ 266.409140] Btrfs loaded, crc32c=crc32c-intel
Scanning for Btrfs filesystems
done.
```

y comprobamos qué tenemos con lsblk:

```
geraylp@ubuntu2004:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                               7:0      0   55M  1 loop  /snap/core18/1880
loop1                               7:1      0  71.3M  1 loop  /snap/lxd/16099
loop2                               7:2      0  67.8M  1 loop  /snap/lxd/22753
loop3                               7:3      0  55.6M  1 loop  /snap/core18/2566
loop4                               7:4      0   48M  1 loop  /snap/snapd/17336
loop5                               7:5      0  63.2M  1 loop  /snap/core20/1623
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
│           ├──vg0-swap          253:1    0    1G  0 lvm    [SWAP]
│           ├──vg0-hogar         253:2    0  500M  0 lvm    /home
│           └─vg0-raiz           253:3    0   8.1G  0 lvm    /
sr0                                 11:0     1  1024M  0 rom
```

Hemos arrancado con un único disco

En este estado, no sería correcto dar servicio al cliente porque no estamos cumpliendo el contrato, el RAID prometido, y además si fallara el disco que queda cagadolahemos.

6. Desde VirtualBox volvemos a crear un disco para volver a montarlo todo como lo teníamos (es hotpluggable)
7. Volvemos a configurar sdb tal y como lo teníamos con fdisk

```

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-4194303, default 2048):
Last sector, +/-sectors or +/-size[K,M,G,T,P] (2048-4194303, default 4194303): +400M

Created a new partition 1 of type 'Linux' and of size 400 MiB.

Command (m for help): n
Partition type
  p   primary (1 primary, 0 extended, 3 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (616448-4194303, default 616448):
Last sector, +/-sectors or +/-size[K,M,G,T,P] (616448-4194303, default 4194303):

Created a new partition 2 of type 'Linux' and of size 1.6 GiB.

Command (m for help): p
Disk /dev/sdb: 2 GiB, 2147483648 bytes, 4194304 sectors
Disk model: VBOX HARDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk Identifier: 0xdd5e061

Device      Boot  Start      End  Sectors  Size Id Type
/dev/sdb1                2048  616447    614400   400M 83 Linux
/dev/sdb2             616448 4194303 3577856   1.6G 83 Linux

Command (m for help): _

```

8. Le instalamos grub al disco

```
sudo grub-install /dev/sdb
```

9. Y volvemos a añadir el disco con sus particiones al RAID

```

sudo mdadm -a /dev/md0 /dev/sdb1
sudo mdadm -a /dev/md1 /dev/sdb2
watch -n 1 cat /proc/mdstat

```

10. lsblk para comprobar otra vez que está bien

Preguntas

- **Qué archivo miramos**
 - /proc/mdstat
- **Leer mensajes del kernel**
 - dmesg
- **Eliminar un disco, marcarlo defectuoso, volverlo a añadir a un raid**
 - Marcar defectuoso: `sudo mdadm --manage --set-faulty /dev/md0 /dev/sdb1`
 - Eliminar: `sudo mdadm -r /dev/md0 /dev/sdb1`
 - Volverlo a añadir: `sudo mdadm -a /dev/md0 /dev/sdb1`
- **Protocolo cuando falla un raid**
 - Fallo software: eliminar (`mdadm -r`) y volverlo a añadir (`mdadm -a`)
 - Fallo hardware, falla boot: nos llevará a `initramfs`. Desde ahí, siempre lo primero mirar: `lvm`, `dmesg`, `mdstat`... Forzar activación de los discos (`mdadm -R`) y continuar con el arranque (`ctrl+d`) con lo que tengamos
- **Qué significa `initramfs`**

- All 2.6 Linux kernels contain a gzipped "cpio" format archive, which is extracted into rootfs when the kernel boots up. After extracting, the kernel checks to see if rootfs contains a file "init", and if so it executes it as PID 1. If found, this init process is responsible for bringing the system the rest of the way up, including locating and mounting the real root device (if any). If rootfs does not contain an init program after the embedded cpio archive is extracted into it, the kernel will fall through to the older code to locate and mount a root partition, then exec some variant of /sbin/init out of that.
 - initramfs is the solution introduced for the 2.6 Linux kernel series. The idea is that there's a lot of initialisation magic done in the kernel that could be just as easily done in userspace.
- **Donde encontrar y como interpretar mdstat**
 - Ubicación: /proc/mdstat
 - Significado: multidevice status
 - La parte importante es el estado de los discos: U para up, _ para caído
 - **Que hacer en initramfs cuando falla un raid para que el sistema inicie igualmente**
 - Forzar activación de los discos con mdadm -R /dev/mdx (los que tengamos que hacer, mirar en mdstat) y CTRL+D para continuar con el boot
 - **Seguir dando acceso o no**
 - No, estaríamos incumpliendo el contrato

P3L2 - Zabbix

Instalación UbuntuServer

1. **Instalación siguiendo las instrucciones de la página de instalación, tiene un selector múltiple que nos da las instrucciones según la versión que queremos, el sistema en el que lo instalamos...**

1. Instalar repo

```
wget [URL]
sudo dpkg -i zabbix....
sudo apt update
```

2. Instalar servidor, interfaz, agente...

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

3. Crear BD

```
sudo mysql -uroot -p
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> create user zabbix@localhost identified by 'practicass,ISE';
mysql> grant all privileges on zabbix.* to zabbix@localhost;
mysql> quit;
```

4. Importar esquema y los datos al servidor de zabbix

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

Comprobar que es correcto intentando ejecutar otra vez, si nos dice que ya existe es que bien

2. Configuración: /etc/zabbix/zabbix_server.conf

1. Contraseña de la BD: en el archivo de conf, buscar DBPassword, descomentar y poner practicas,ISE
2. PHP para interfaz zabbix
 1. Ir a /etc/zabbix/apache.conf y descomentar las dos lineas de date.timezone. Poner Madrid en vez de Riga.
 2. Reiniciar y habilitar servicios


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

Podemos comprobar en el log: /var/log/zabbix/zabbix_server.log

3. Configuración del servidor usando la guía de inicio rápido

1. Abrir en un navegador: <http://192.168.56.105/zabbix/setup.php>
2. Configurar la timezone

Check of pre-requisites

 Time zone for PHP is not set (configuration parameter "date.timezone").			
	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"	unknown		Fail
PHP databases support	MySQL		OK

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

Para arreglar la zona horaria nos vamos a `/etc/php/7.4/apache2/php.ini` (Cuidado con la version de php, poned la vuestra.) y descomentamos la linea `date.timezone(;` comenta en php) y ponemos "Europe/Madrid":

Reiniciar apache con `sudo systemctl restart apache`

3. Configuración BD

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](#)

La contraseña de siempre: practicas,ISE

4. Configuración puerto del servidor

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	localhost
Port	10051
Name	Test

5. Logear con user: Admin, pass: zabbix

4. Configurar el agente del servidor

1. En /etc/zabbix/zabbix_agentd.conf buscar /server: es donde vamos a ver las IPs que acepta el agente. Añadimos la ip de nuestra máquina y la de rocky
2. Crear nuevo host cambiando el hostname
3. Reiniciar el servicio sudo systemctl restart zabbix-agent

Instalación Rocky

1. Instalación siguiendo instrucciones de la página

1. Repositorio

```
sudo rpm [URL]
```

2. Agente

```
sudo dnf install zabbix-agent
```

2. Configuración del agente

```
sudo vi /etc/zabbix/zabbix_agentd.conf
Server=192.168.56.105 #Ip del servidor(Ubuntu) Hostname=Host Rocky Linux
```

Añadiendo la ip de ubuntu y la de rocky para poder hacer zabbix-get

Luego reiniciamos y habilitamos

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

3. Creamos un host

1. Vamos al navegador <http://192.168.56.105/zabbix/> y Configuration>Hosts>Create host
2. En interfaz del agente ponemos la ip de rocky

Type	IP address	DNS name	Connect to	Port
Agent	192.168.56.110		IP	10050

3. IMPORTANTE habilitar los puertos en el cortafuegos

```
sudo firewall-cmd --add-port=10050/tcp --permanent #Permite a zabbix-server
sudo firewall-cmd --add-port=22/tcp --permanent #Permite a ssh
sudo firewall-cmd --add-port=80/tcp --permanent #Permite a http
sudo firewall-cmd --reload
```

Monitorización

Para monitorizar el ssh y http tenemos 2 opciones: usar Templates o añadir los items a mano. Ambas formas son válidas siempre y cuando en los templates modifiquemos el puerto correspondiente y entendamos como funcionan.

Templates:

1. Configuration>Hosts, elegir el host que hemos configurado antes e ir a templates. Añadir las templates de SSH y HTTP

Link new templates
Template App SSH Service
Template App HTTP Service

2. Configuration>Templates y buscamos las que queremos modificar (ssh, http). Columna Items

Templates

Host groups: type here to search [Select] Tags: And/Or

Linked templates: type here to search [Select] tag Add

Name: **ssh**

Apply Reset

Name	Hosts	Applications	Items	Triggers	Graphs
Template App SSH Service	Hosts 1	Applications 1	Items 1	Triggers 1	Graphs

0 selected Export Mass update Delete Delete and clear

igual para http

3. Nombre del item

Items

All templates / Template App SSH Service Applications 1 **Items 1** Triggers 1 Graphs Screens Discovery rules Web scenarios

Host groups: type here to search [Select] Type

Hosts: Template App SSH ... [X] [Select] Update interval

Application: [Select]

Name: **SSH service is running**

Key: net.tcp.service[ssh]

Subfilter affects only filtered data

Name	Triggers	Key
SSH service is running	Triggers 1	net.tcp.service[ssh]

4. Añadir puertos

1. net.tcp.service[ssh,,22]
2. net.tcp.service[http,,80]

Items

All templates / Template App SSH Service Applications 1 **Items 1** Triggers 1 Graphs Screens Discovery rules Web scenarios

Item Preprocessing

Name: SSH service is running

Type: Simple check

Key: **net.tcp.service[ssh,,22]** [Select]

User name: []

Password: []

Type of information: Numeric (unsigned)

A mano:

1. Configuration>Hosts, elegimos nuestro Host y vamos a Items.
2. Creamos un nuevo item

Item
Preprocessing

* Name
SSH MONITORING

Type
Zabbix agent

* Key
net.tcp.service[ssh,,22]
Select

* Host interface
127.0.0.1 : 10050

Type of information
Numeric (unsigned)

* Name
HTTP MONITORING

Type
Zabbix agent

* Key
net.tcp.service[http,,80]
Select

* Host interface
127.0.0.1 : 10050

Type of information
Numeric (unsigned)

Name ▲	Triggers	Key	Interval	History	Trends	Type
HTTP MONITORING		net.tcp.service[http,,80]	1m	90d	365d	Zabbix agent
SSH MONITORING		net.tcp.service[ssh,,22]	1m	90d	365d	Zabbix agent

Comprobar la conexión

Podemos tirar un servicio (sudo systemctl stop ssh) y esperar a ver qué pasa en Monitoring>Problems. También cuenta en history con una gráfica del uptime.

Preguntas

- **Puertos por defecto zabbix**
 - zabbix: 10051
 - ssh: 22
 - http: 80
- **Cómo utilizar zabbix con consola/sin UI**
 - zabbix-get
- **Qué problema tiene rocky con zabbix**
 - Problema: Tenemos que habilitar los puertos
 - Solución:

```
sudo firewall-cmd --add-port=10050/tcp --permanent #Permite a zabbix-server
sudo firewall-cmd --add-port=22/tcp --permanent #Permite a ssh
sudo firewall-cmd --add-port=80/tcp --permanent #Permite a http
sudo firewall-cmd --reload
```

- **Cortafuegos con el agente de zabbix**
 - Tenemos que habilitar los puertos de zabbix, ssh y http con --add-port y luego recargar el cortafuegos.
- **Cómo monitorizar un servicio en zabbix**
 - UI: Monitoring>Latest data
 - Consola: zabbix-get
- **Dónde están los archivos de conf de zabbix**

- Config: /etc/zabbix/zabbix_server.conf
- Apache: /etc/zabbix/apache.conf
- Agente: /etc/zabbix/zabbix_agentd.conf
- Log: /var/log/zabbix/zabbix_server.log
- **Proxy de zabbix (zabbixproxy)**
 - Zabbix proxy is a process that may collect monitoring data from one or more monitored devices and send the information to the Zabbix server, essentially working on behalf of the server. All collected data is buffered locally and then transferred to the Zabbix server the proxy belongs to.

Deploying a proxy is optional, but may be very beneficial to distribute the load of a single Zabbix server. If only proxies collect data, processing on the server becomes less CPU and disk I/O hungry.
- **Usuario y contraseña de frontend de zabbix**
 - User: Admin
 - Pass: zabbix
- **Que se modifica en la configuración de zabbix**
 - La timezone
- **Relación de LAMP y zabbix**
 - Apache y MySQL
- **Cosas que hacer en el frontend de zabbix una vez instalado**
 - Añadir los items que queremos monitorizar, en este caso ssh y http. Podemos hacerlo a mano o usando templates.

P3L3 Ansible

Instalación y conexión

1. Instalación según la docu

```
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

2. Añadir hosts en /etc/ansible/hosts

```
#[webservers]
#alpha.example.org
#beta.example.org
192.168.56.105 ansible_user=yeraulp
192.168.56.110 ansible_user=yeraulp
```

Añadimos la ip de ubuntu y rocky, además indicamos el usuario que debe usar ansible para logearse

3. Generar y añadir las claves publicas de ambos servidores para que ansible pueda conectarse por ssh.

```
#En ubuntu
ssh-keygen
ssh-copy-id 192.168.56.105 #Añadimos la key de ubuntu al propio ubuntu
ssh-copy-id 192.168.56.110 # Añadimos la key de ubuntu a rocky
```

4. Comprobar que ansible se conecta a los servidores con ping

```
ansible all -m ping
```

Creación de un inventario

Los inventarios nos permiten agrupar hosts en grupos.

1. Creamos el fichero:

```
vi inventory.yaml

virtualmachines:
  hosts:
    ubuntu2004:
      ansible_host: 192.168.56.105
    rocky:
      ansible_host: 192.168.56.110
```

2. Podemos hacer ping de nuevo con el nuevo grupo

```
ansible virtualmachines -m ping -i inventory.yaml
```

Creación de un playbook

Los playbook son listas de tareas que ansible ejecuta de una en una en los servidores que especifiquemos.

1. Creamos el fichero:

```
vi playbook.yaml

- name: Libro de prueba #Nombre del libro
  hosts: virtualmachines
  tasks: #Lista de tareas
    - name: Ping a los hosts #1ªTarea
      ansible.builtin.ping:
    - name: Imprime un mensaje #2ªTarea
      ansible.builtin.debug: msg: Hello world
```

2. Ejecutando un playbook:

```
ansible-playbook -i inventory.yaml playbook.yaml
```

Preguntas

- **Ansible y su relación con ssh**

- Ansible's main goals are simplicity and ease-of-use. It also has a strong focus on security and reliability, featuring a minimum of moving parts, usage of OpenSSH for transport (with other transports and pull modes as alternatives)
- Utiliza ssh para conectarse a los servidores. Necesita para ello generar llaves públicas

- **Qué hace comando ping de ansible**

- Hace ping a los servidores que hayamos añadido en el archivo de hosts

- **Cómo se denominan los archivos de ansible y qué lenguaje utilizan**

- Playbooks, YAML

- **Cómo instalar paquetes mediante ansible**

- Mediante playbook (instala sysstat, httpd, mariadb-server):

```
---
- hosts: all
  tasks:
    - name: Package installation
      dnf:
        name:
          - sysstat
          - httpd
          - mariadb-server
        state: latest
```

- Mediante terminal:

```
ansible all --user tux --become --module-name dnf -a 'name=sysstat
state=latest'
```

- **Cómo se llama no tener un agente instalado con ansible**

- Agentless architecture

- **Archivos a modificar para que ansible funcione**

- /etc/ansible/hosts

- **Cómo comprobar el estado del servicio http mediante un playbook**

- ```
- name: checking service status
 hosts: www.linuxfoundation.org
 tasks:
 - name: checking service status
 command: systemctl status httpd
 register: result
 ignore_errors: yes
 - name: showing report
 debug:
 var: result
```

- **Cómo instalar httpd en rocky en un playbook**

```

o ---
- hosts: all
 tasks:
 - name: Package installation
 dnf:
 name:
 - httpd
 state: latest

```

## Comandos documento de la práctica y otras preguntas

- hddtemp, lm-sensors, lspci, lsusb, lshw -- Monitorización hardware
- dmesg -- Mensajes del kernel
- strace -- traza de llamadas al sistema, útil para cosas que no guardan los logs (/var/log)

## Preguntas

- **Cómo ver la salida de los timer, cómo ejecutarlos**
  - Ver timers activos: systemctl list-timers
  - Ejecutar: journalctl
- **Cómo automatizar un script con systemd**
  - Podemos automatizar la ejecución de un script en systemd definiendo un timer dentro del directorio /etc/systemd/system/ que se encarga de gestionar un servicio. Por tanto, hemos de crear dos archivos: mon\_raid.timer y mon\_raid.service
- **Qué hacer no route to host**
  - Comprobar que todos los servicios están corriendo con systemctl status (y si no, arreglarlos)
  - Comprobar los puertos
    - Comprobar que el firewall no está interfiriendo
- **Szk y para qué se usa**
  - //Será que copiamos mal el nombre porque no encuentro nada de nada
- **lm-sensors y por qué no lo hemos usado**
  - Porque monitoriza aspectos del hardware. No tiene sentido utilizarlo en una máquina virtual, porque monitorizará hardware simulado.
- **Cómo hacer para encontrar un fichero**
  - find
    - Ej. uso: copiar todos los archivos pdf en un directorio

```
find /home/alberto/docs -name '*pdf' -exec cp {} ~/PDFs \;
```

- **Cómo ver la última vez que inició sesión un user sospechoso, donde estan los logs**

```
grep "Failed password" /var/log/auth.log
```

- **Cómo se mira la última vez que se conectó un usuario**



```
grep "[nombre del usuario]" /var/log/auth.log
```

//Habría q afinar un poco más esto para q solo devolviera el último pero jemapelle barbara (me la pela una barbaridad)

- **Monitorización en la disponibilidad, como evitar cosas malas**
- **Problema de usar muchos nodos y soluciones**
- **Que hacer si se cae el servicio de apache en ubuntu server**