

## LECCIÓN 2

---

Nos conectamos a ubuntu por ejemplo. Directorio `/proc` es un directorio que monta el kernel de linux y no se corresponde con ficheros que se guarden en un dispositivo de almacenamiento permanente. No se guarda en memoria; se crea cada vez que se arranca el sistema. Si nos metemos en `/proc`, veremos directorios con nombre numeritos. Esos numeritos son identificadores de procesos y si nos metemos dentro de ellos podemos ver información de dichos procesos. Un contenido interesante es el archivo: **cpuinfo** (info del sistema) y otra cosa que cae en el examen es **loadavg** que nos da las medias de uso de cpu durante el último minuto, 5 minutos y 10 minutos (según la documentación de average) y a/b que indica que hay ... (no me he enterado bien)

Los primeros 3 números de los 5 que aparecen en `loadavg` son el número medio de tareas que querían entrar a la cpu para ejecutarse en los últimos 1/5/10 minutos.

Por ejemplo: en caso de 1 core y fijándonos en el campo de 1 minuto: si nos sale 0.5 quiere decir que el 50% de la cpu ha estado ocupada en el último minuto.

Hacemos `sudo apt install stress`. Stress permite crear procesos para saturar la cpu.

Hacemos `stress --cpu 8 --io 4 --vm 2` (en una terminal de la maquina)

```
watch -n 1 cat /proc/loadavg
```

 (en otra terminal de la máquina) que saca cada segundo la salida de ese comando

Hacemos `top`: - Tasks = hebras (total, dormidas, corriendo, paradas o zombies (estas dos últimas casi siempre están a 0, si no estuvieran a 0 es porque tendríamos procesos que seguramente no estén liberando bien los recursos.)) - %CPU = espacio de ejecuciones de usuario, del sistema, nice (ni: procesos priorizados), idle (vacío), wa (waiting esperando a una operación de entrada/salida) - **Uso de memoria:** MiB Men cómo estamos usando la memoria principal (total, libre, usada, usadas en buffer y cache)- EL INDICADOR DE LA FILA DE ABAJO, **avail Mem** es de memoria principal, lo que pasa es que se baja a la línea de abajo. - **Uso de memoria virtual** MiB Swap (no es bueno que el indicador de memoria virtual, used no sea 0. Tiene que estar a cero preferiblemente.)

Haciendo `free` nos salen las mismas cosas de la cabecera de `top`. Tanto `top` como `free` sacan datos de `meminfo`, `vmstat`. Haciendo `uptime` nos muestra los indicadores de tiempo de `top` (arriba del todo) pero en otra unidad (seg).

## Cron

Es un proceso, un servicio que podemos gestionar con `systemctl` y es de los primeros que arranca y ejecuta un archivo

El archivo de configuración de cron es:

```
cd /etc/ more crontab
```

"\*" significa todas las horas/dias/meses...

17 \* \* \* \* username comando : ejecuta el comando en modo username en el minuto 17 de todas las horas de todos los días del mes de todos los meses de l año todos los días de la semana. Es recomendable pasar siempre los path completos.

Nosotros vamos a crear nuestro propio cron en el espacio de un usuario no privilegiado:

crontab -e (para editarlo) choose : nano Es lo mismo que el cron maestro en cuanto a formato pero no me da opción a elegir usuario, solo me da mi usuario.

Hay un programa que se llama logger que manda cosas al syslog:

```
which logger
echo "Prueba logger" | logger
```

Ponemos en crontab:

```
* * * * * echo "Practicas ISE" | /usr/bin/logger -t "ISE"
```

Comprobamos que se ha hecho:

```
tail -f /var/log/syslog
```

## Zabbix

Monitor de proposito general que recava información de métricas del sistema y nos las almacena en una base de datos y nos permite visualizarlas de forma gráfica.

Instalamos zabbix en ubuntu (zabbix 5): nos vamos a la página web oficial de zabbix y lo instalamos:

Vamos a instalar en ubuntu el **zabbix server** y luego vamos a monitorizar una sola máquina (se hace en verdad con más). En las máquinas instalamos los agentes. El server mediante una operación **pull** va pidiendo a los agentes de las máquinas que le vayan pasando métricas. Pero, ¿cómo lo hace? el server tiene una base de datos y una tabla donde tiene definiidos los distintos hosts (máquinas que vamos a monitorizar) los valores concretos de la información los va guardando en tablas internas de tal forma que con el paso del tiempo pues tiene información almacenada y con ella puede sacar una gráfica. Presenta el server la información en forma de página web. Para mostrar la gráfica final necesita, apache, php y mysql (LAMP). Tenemos que asegurarnos que no hay problemas en los puertos (firewall controlarlo). **get** nos ayuda a depurar la comunicación entre el servidor y el agente.

ANTES DE INSTALAR TODO ESTO INSTALAR LAMP (y en vez de mariadb, instalar mysql)

Vamos a installation --> instalar por paquetes (from packages) --> debian/ubuntu --> vamos al enlace [installation.instructions](#) --> nos aparecerá una guía donde nos explica lo que vamos a hacer --> copiamos los comandos del paso 2 uno por uno.

Nos conectamos a `http://direccion_maquina/zabbix` y rellenamos todo el formulario.

Por último vamos a quickstart. El login por defecto es admin y la contraseña por defecto es zabbix. Nos logeamos y nos aparecerá la consola oficial de zabbix.

Vamos a la parte de configuración --> host group (que no son más que agrupaciones de hosts)

En configuración --> templates: Los templates permiten crear plantillas con métricas, alarmas, etc que queramos meter en una máquina.

En configuración --> Hosts. Tenemos que crear un host. Podemos ver también los templates del host. Los templates derivan en la creación de Applications. Si le damos a applications vemos las aplicaciones que se han creado y a la derecha de cada aplicación el número de items (métricas). Cuando veamos una metrica (item) que tiene asociado un trigger, eso es una alarma (trigger == alarma). Todas las alarmas van a salir en la pantalla de la consola principal para que el administrador sea consciente de ellas. Antes de crear el host:

- En Rocky instalamos el agente (zabbix 5). Para ello, hacemos:

- La instalación en el mismo enlace de antes, pero en vez de elegir ubuntu, elegimos rocky (red hat...) y seguimos todos los pasos haciendo todos los comandos (como root) que nos diga la página web.

- Una vez instalado zabbix en rocky activamos:

```
todo en modo root: systemctl enable zabbix-agent
systemctl start zabbix-agent
```

- Tenemos que abrir el puerto:

```
firewall-cmd --list-all
firewall-cmd --get-services
firewall-cmd --add-service=zabbix-agent
firewall-cmd --runtime-to-permanent
```

- Zabbix solo permite consultar al localhost (server =localhost). Desde la terminal de rocky

```
- nano -w zabbix_agentd
- Linea Server = 127.0.0.1, 192.168.56.0/24
- Guardamos
- Reiniciamos el servicio porque he tocado la configuración: systemctl
restart zabi
```

Una vez instalado el agente en rocky:

- Comprobamos que tenemos todo listo. Para ello, hacemos una petición

Si el servidor quiere pedirle a rocky su carga de cpu en los últimos 15

```
minutos tiene que comunicarse con su agente y entonces el agente le devuelve un dato: el servidor manda por ejemplo system.cpu.load<cpu,...>
```

Desde ubuntu instalamos zabbix-agent: `sudo apt install zabbix-agent`. Si nos sale un error de que se está ejecutando un proceso que todavía no ha terminado bloquea la instalación hay que hacer una movida rara para desactivar el proceso: pedir a Nour que la he visto apuntando.

Una vez arreglado instalamos zabbix-get.

Hacemos `zabbix_ tabulador tabulador`.

`man zabbix_get`

`copiamos zabbix_get -s ip -p 10050 -k "system.cpu.load[,avg5]"`

Si nos sale 0, pues bueno, no pasa nada pero es raro, pero si hubiera algún problema nos saldría un error.

Ya tenemos comunicación.

Nos vamos a host --> create host --> nombre rocky --> group linux servers --> ip --> puerto okk , resto okk

Vamos a añadir al host unos templates:

- Template OS Linux by Zabbix agent
- Template Module Linux CPU by Zabbix agent
- Template Module Linux memory by Zabbix agent
- Template App SSH Service
- Template App Apache by HTTP

Nos puede salir un error porque puede ser que algún template de linux ya tenga alguno de los otros de forma intrínseca. Quitamos el de linux (Template OS Linux by Zabbix agent).

Si queremos consultar el valor de un item, nos vamos a monitoring --> latest data. Filtramos por memoria por ejemplo (primero le indicamos el host). Podemos incluso verlo gráficamente.

Para no tener que estar esperando a que la gráfica se actualice, nos vamos a configuración --> hosts -> item --> item\_que\_queremos --> entramos dentro y le damos a execute now (botoncillo de abajo).

Desde rocky, si queremos como item sshd:

`systemctl stop sshd` Vamos al host y le damos a execute now veremos que la gráfica decrece drásticamente. Lo levantamos de nuevo: `systemctl start sshd` Volvemos al host y refrescamos gráfica y veremos que crece de nuevo.

## EXAMEN

SE PUEDE TRAER ZABBIX INSTALADO. SE PUEDE TRAER ANSIBLE INSTALADO. (corre ansible desde nuestro anfitrión) SE PUEDE TRAER SSH, LAMP, ... INSTALADO Parte del examen es el quick start Instalar un host. Medir métricas Reiniciar un servicio. Capturas de pantalla de los pasos creando el host. Ubuntu con zabbix instalado y una máquina para monitorizar. Todas las máquinas

--> ansible, zabbix, cron, métricas --> Ansible como ejecutar comando contra varios equipos, ping, apagar máquinas, inventario.