

Instalacion-y-configuration-de-Z...



PruebaAlien



Ingeniería de Servidores



3º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación
Universidad de Granada



BURN
ENERGY DRINK

**GANÁ UN FIAT 500
CON BURN ENERGY**
1 LATA = 1 PARTICIPACIÓN

FIAT 500



ZABBIX



ANSIBLE



BURN
ENERGY DRINK

WUOLAH

Reservados todos los derechos. Queda permitida la impresión en su totalidad.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Índice

Instalación y configuración Zabbix Server en Ubuntu Server	Pg. 1
Instalación Zabbix server, frontend y agent	Pg. 1
Configuración base de datos Zabbix	Pg. 2
Configuración PHP para zabbix frontend	Pg. 4
Habilitar puerto de escucha	Pg. 5
Configuración del agente	Pg. 5
Instalación y configuración Zabbix Agent en CentOS	Pg. 7
Instalación de Zabbix Agent	Pg. 7
Configuración del agente	Pg. 8
FrontEnd	Pg. 9
Configurar monitorización de los servicios SSH y HTTP	Pg. 12
Ubuntu Server	
Creación del host	Pg. 12
Creación de los templates	Pg. 14
CentOS	
Creación del host	Pg. 15
Creación de los templates	Pg. 16
Monitorización de los servicios SSH y HTTP	Pg. 17
Ubuntu Server	Pg. 17
CentOS	Pg. 18
Instalación y configuración Ansible en Ubuntu Server	Pg. 20
Bibliografía	Pg. 30

Estudiar sin publi es posible.



Compra Wuolah Coins y que nada te distraiga durante el estudio



Instalación y configuración de Zabbix Server en Ubuntu Server

Para la instalación de Zabbix server:

1. Necesitamos descargarnos el repositorio Zabbix con la versión 5.0 y con la distribución Focal para Ubuntu con la versión 20.04, **zabbix-release_5.0-1+focal_all.deb**

Para ello ejecutamos el siguiente comando:

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

```
ruben@ubuntu-server:~$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zab-
bix-release_5.0-1+focal_all.deb
--2020-12-05 17:43:04-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zab-
bix-
-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
2020-12-05 17:43:06 (206 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]
ruben@ubuntu-server:~$
```

2. Descomprimimos el paquete

Para descomprimirlo usamos el comando **dpkg** con la opción **-i** para instalar el paquete en el directorio de paquetes:

```
root@ubuntu-server:/home/ruben# dpkg -i zabbix-release_5.0-1+focal_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 75219 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) ...
root@ubuntu-server:/home/ruben#
```

3. Actualizamos la lista de paquetes con **apt update**

```
root@ubuntu-server:~# 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://es.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:4 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4930 B]
Get:5 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Get:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1194 B]
Get:7 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [3216 B]
Fetched 333 kB in 47s (7062 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
124 packages can be upgraded. Run 'apt list --upgradable' to see them.
root@ubuntu-server:~# _
```

4. Y por último instalamos los servicios de **zabbix** (server-mysql, frontend-php, apache-conf y el agente)

- **zabbix-server-mysql** (soporte para MySQL, zabbix server)
- **zabbix-frontend-php** (frontend)
- **zabbix-apache-conf**
- **zabbix-agent** (para que podamos monitorizar el servidor desde la propia maquina)

```
apt-get install zabbix-server-mysql zabbix-frontend-php  
zabbix-agent
```

Configuración base de datos Zabbix Server

5. A continuación, entramos a la base de datos MySQL como root y creamos la base de datos para Zabbix llamada **zabbix**, también habrá que crear un usuario para zabbix y darle todos los privilegios, en este caso el usuario es **zabbix** y la contraseña es igual al del servidor **practicas,ISE**.

```
root@ubuntu-server:/home/ruben# mysql -uroot -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 8  
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)  
  
Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.  
  
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;  
ERROR 1273 (HY000): Unknown collation: 'utf8_bin'  
mysql> create database zabbix character set utf8 collate utf8_bin;  
Query OK, 1 row affected, 2 warnings (0.27 sec)  
  
mysql> create user zabbix@localhost identified by 'practicas,ISE';  
Query OK, 0 rows affected (0.26 sec)  
  
mysql> grant all privileges on zabbix.* to zabbix@localhost;  
Query OK, 0 rows affected (0.36 sec)  
  
mysql> quit;  
Bye  
root@ubuntu-server:/home/ruben# _
```



GANAR UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



6. Importamos los esquemas y datos, nos pedirá la contraseña de usuario creado, esto puede tardar minutos, puesto que importa muchas tablas y muy importante no puede ser interrumpido, puesto que si se interrumpe, no funcionaría bien zabbix y nos daría errores.

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

```
root@ubuntu-server:/home/ruben# zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql -uzabbix  
ix -p zabbix  
Enter password:
```

7. Configuramos la base de datos en zabbix server en **/etc/zabbix/zabbix_server.conf**, como el usuario y el nombre de la base de datos son igual al que trae por defecto cuando instalamos zabbix, solo tenemos que poner la contraseña que le hemos puesto al usuario **zabbix**, en mi caso es **practicas,ISE**

```
### 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=practicas,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" 860L, 21596C written  
root@ubuntu-server:/home/ruben# _
```

Configuración PHP para zabbix frontend

8. Configuramos PHP para zabbix frontend en **/etc/zabbix/apache.conf** y solo descomentamos la zona horaria de la región, en mi caso como estoy en España cambio **Riga** que es el que trae por defecto, por **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
</Directory>
"/etc/zabbix/apache.conf" 68L, 1665C written
root@ubuntu-server:/home/ruben#
```

9. Despues una vez configurado, reiniciamos y habilitamos los servicios **zabbix-server**, **zabbix-agent** y **apache2**

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

```

root@ubuntu-server:/home/ruben# systemctl restart zabbix-server zabbix-agent apache2
root@ubuntu-server:/home/ruben# 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 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/sys-
tem/zabbix-server.service.
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /lib/systemd/sys-
tem/zabbix-agent.service.
root@ubuntu-server:/home/ruben#

```

Habilitar puerto de escucha

10. Habilitamos el puerto de escucha que usa por defecto zabbix con el comando **ufw allow 10051/tcp**, si no se hace esto no funcionara el frontend, ya que utiliza el puerto **10051** por defecto, ya que Ubuntu server el cortafuegos bloquea por defecto todos los puertos

```

root@ubuntu-server:/home/ruben# ufw status
Status: active

To                         Action      From
--                         ----      --
22022                       ALLOW      Anywhere
80                          ALLOW      Anywhere
443                          ALLOW      Anywhere
10050/tcp                   ALLOW      Anywhere
22022 (v6)                  ALLOW      Anywhere (v6)
80 (v6)                     ALLOW      Anywhere (v6)
443 (v6)                    ALLOW      Anywhere (v6)
10050/tcp (v6)              ALLOW      Anywhere (v6)

root@ubuntu-server:/home/ruben# ufw allow 10051/tcp
Rule added
Rule added (v6)
root@ubuntu-server:/home/ruben# ufw status
Status: active

To                         Action      From
--                         ----      --
22022                       ALLOW      Anywhere
80                          ALLOW      Anywhere
443                          ALLOW      Anywhere
10050/tcp                  ALLOW      Anywhere
10051/tcp                  ALLOW      Anywhere
22022 (v6)                 ALLOW      Anywhere (v6)
80 (v6)                     ALLOW      Anywhere (v6)
443 (v6)                    ALLOW      Anywhere (v6)
10050/tcp (v6)              ALLOW      Anywhere (v6)
10051/tcp (v6)              ALLOW      Anywhere (v6)

root@ubuntu-server:/home/ruben# _

```

Configuración del agente

11. Configuramos el agente accediendo al archivo **/etc/zabbix/zabbix_agentd.conf**, aquí le indicamos la IP del servidor, por defecto tiene la del localhost y también en el **serverActive** hay que indicarle la IP del servidor. Esto nos servirá para monitorizar fuera del servidor de Ubuntu, en mi caso desde CentOS

```

### 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

### Option: ServerActive
#      List of comma delimited IP:port (or DNS name:port) pairs of Zabbix servers and Zabbix proxies for active checks.
#      If port is not specified, default port is used.
#      IPv6 addresses must be enclosed in square brackets if port for that host is specified.
#      If port is not specified, square brackets for IPv6 addresses are optional.
#      If this parameter is not specified, active checks are disabled.
#      Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.56.105_

```

12. Reiniciamos los servicios del **zabbix-agent** y comprobamos el estado de los servicios **zabbix-server**, **zabbix-agent** y **apache2**

```

● zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-12-05 21:25:51 UTC; 33min ago
     Main PID: 18247 (zabbix_server)
        Tasks: 38 (limit: 2282)
       Memory: 68.5M
      CGroup: /system.slice/zabbix-server.service
              └─18247 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
                  ├─18262 /usr/sbin/zabbix_server: configuration syncer [synced configuration in 0.14596s]
                  ├─18268 /usr/sbin/zabbix_server: housekeeper [deleted 0 hist/trends, 0 items/triggers, 0 events]
                  ├─18269 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.000000s]
                  ├─18271 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000416 sec, idle 5s]
                  ├─18273 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.000415 sec, idle 5s]
                  ├─18274 /usr/sbin/zabbix_server: history syncer #1 [processed 0 values, 0 triggers in 0.000000s]
                  ├─18275 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0 triggers in 0.000000s]
                  ├─18276 /usr/sbin/zabbix_server: history syncer #3 [processed 1 values, 1 triggers in 0.000000s]
                  ├─18280 /usr/sbin/zabbix_server: history syncer #4 [processed 0 values, 0 triggers in 0.000000s]
                  ├─18281 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.000553 sec]
                  ├─18282 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.000000s]
                  ├─18284 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000020 sec, idle 5s]
                  ├─18285 /usr/sbin/zabbix_server: task manager [processed 0 tasks(s) in 0.000244 sec, idle 5s]
                  ├─18289 /usr/sbin/zabbix_server: poller #1 [got 0 values in 0.000042 sec, idle 5 sec]
                  ├─18292 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000024 sec, idle 5 sec]
                  ├─18293 /usr/sbin/zabbix_server: poller #3 [got 1 values in 0.000067 sec, idle 5 sec]
                  ├─18294 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000031 sec, idle 5 sec]
                  ├─18295 /usr/sbin/zabbix_server: poller #5 [got 0 values in 0.000022 sec, idle 5 sec]
                  ├─18296 /usr/sbin/zabbix_server: unreachable poller #1 [got 0 values in 0.000037 sec, waiting f>
                  ├─18297 /usr/sbin/zabbix_server: trapper #1 [processed data in 0.000000 sec, waiting f>
                  ├─18298 /usr/sbin/zabbix_server: trapper #2 [processed data in 0.000000 sec, waiting f>
                  ├─18299 /usr/sbin/zabbix_server: trapper #3 [processed data in 0.038637 sec, waiting f>
                  ├─18300 /usr/sbin/zabbix_server: trapper #4 [processed data in 0.000000 sec, waiting f>
                  ├─18301 /usr/sbin/zabbix_server: trapper #5 [processed data in 0.000000 sec, waiting f>
                  ├─18303 /usr/sbin/zabbix_server: icmp pinger #1 [got 0 values in 0.000034 sec, idle 5 sec]
                  ├─18305 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.000000s]
                  ├─18310 /usr/sbin/zabbix_server: alerter #1 started
                  ├─18315 /usr/sbin/zabbix_server: alerter #2 started

```

lines 1-36



GANAR UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



```

● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-12-05 21:57:38 UTC; 1min 26s ago
     Process: 18894 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
    Main PID: 18911 (zabbix_agentd)
      Tasks: 6 (limit: 2282)
        Memory: 5.4M
       CGroup: /system.slice/zabbix-agent.service
               ├─18911 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
               ├─18912 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
               ├─18913 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
               ├─18914 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
               ├─18915 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
               └─18916 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

Dec 05 21:57:37 ubuntu-server systemd[1]: Starting Zabbix Agent...
Dec 05 21:57:38 ubuntu-server systemd[1]: zabbix-agent.service: Can't open PID file /run/zabbix/zab
Dec 05 21:57:38 ubuntu-server systemd[1]: Started Zabbix Agent.

● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-12-05 21:26:03 UTC; 33min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 18263 (apache2)
      Tasks: 7 (limit: 2282)
        Memory: 20.1M
       CGroup: /system.slice/apache2.service
               ├─18263 /usr/sbin/apache2 -k start
               ├─18264 /usr/sbin/apache2 -k start
               ├─18265 /usr/sbin/apache2 -k start
               ├─18266 /usr/sbin/apache2 -k start
               ├─18267 /usr/sbin/apache2 -k start
               ├─18272 /usr/sbin/apache2 -k start
               └─18686 /usr/sbin/apache2 -k start

lines 51-86

```

Como vemos no necesitamos iniciar ningún servicio, puesto que están arrancados.

Instalación y configuración de Zabbix CentOS

Para la instalación de Zabbix Agent:

1. Descargamos el paquete zabbix del repositorio

```
rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
```

```
[root@localhost ruben]# rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
Recuperando https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
advertencia: /var/tmp/rpm-tmp.XU190M: EncabezadoRSA SHA512 Signature, ID de clave a14fe591: NOKEY
Verifying... ###### [100%]
Preparando... ###### [100%]
Actualizando / instalando...
 1:zabbix-release-5.0-1.el8 ###### [100%]
[root@localhost ruben]#
```

2. Instalamos el agente únicamente, puesto que es lo que nos pide el ejercicio usando **yum install zabbix-agent**

```
Instalado:
  zabbix-agent-5.0.6-1.el8.x86_64

¡Listo!
[root@localhost ruben]# _
```

Configuración del agente

3. Antes de configurar el agente, tenemos que habilitar el puerto que por defecto es **10050**, puesto que en CentOS todos los puertos están cerrados por el cortafuegos, pero como ya sabemos en centOS hay que activarlo ahora y de forma permanente en 2 comandos, ya que uno lo aplica de forma instantánea, pero temporalmente, y la otra se aplica al reiniciar la maquina

```
[root@localhost ruben]# firewall-cmd --add-port=10050/tcp --permanent  
success  
[root@localhost ruben]# firewall-cmd --add-port=10050/tcp  
success
```

```
public (active)  
target: default  
icmp-block-inversion: no  
interfaces: enp0s3 enp0s8  
sources:  
services: cockpit dhcpv6-client ssh  
ports: 22022/tcp 80/tcp 443/tcp 10050/tcp  
protocols:  
masquerade: no  
forward-ports:  
source-ports:  
icmp-blocks:  
rich rules:
```

Y como se puede ver esta ya activada

4. Una vez habilitado el puerto configuramos el agente **/etc/zabbix/zabbix_agentd.conf**, indicando el servidor zabbix, que esta en Ubuntu Server con la IP **192.168.56.105**

```
### 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
```

```
### Option: ServerActive  
#      List of comma delimited IP:port (or DNS name:port) pairs of Zabbix servers and Zabbix proxies for active checks.  
#      If port is not specified, default port is used.  
#      IPv6 addresses must be enclosed in square brackets if port for that host is specified.  
#      If port is not specified, square brackets for IPv6 addresses are optional.  
#      If this parameter is not specified, active checks are disabled.  
#      Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]  
  
# Mandatory: no  
# Default:  
# ServerActive=  
  
ServerActive=192.168.56.105
```

A parte le voy a cambiar el nombre del hostname, para distinguir entre el agente de centos y el de Ubuntu server

```
Hostname=Zabbix agent-CentOS
### Option: HostnameItem
# Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
"/etc/zabbix/zabbix_agentd.conf" 509L, 15117C written
[root@localhost ruben]#
```

5. Habilitamos las políticas de SELinux **sudo yum install policycoreutils-python-utils.noarch**, pero como ya se han habilitado anteriormente, en la practica 2, no será necesario.
6. Comprobamos que esta activado **zabbix-agent**, como no está activado lo activamos

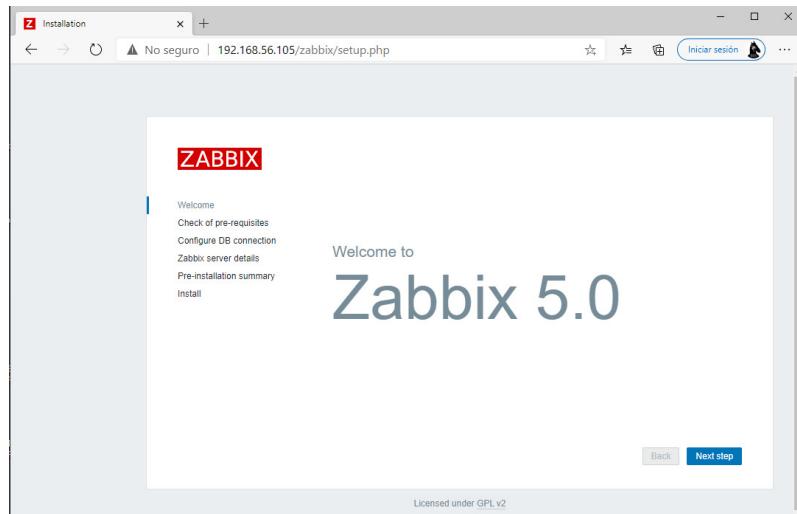
```
[root@localhost ruben]# systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
  Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; disabled; vendor preset: disabled)
  Active: inactive (dead)
[root@localhost ruben]# systemctl start zabbix-agent
[root@localhost ruben]# systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
  Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; disabled; vendor preset: disabled)
  Active: active (running) since Sun 2020-12-06 15:29:26 EST; 36s ago
    Process: 26643 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
   Main PID: 26645 (zabbix_agentd)
     Tasks: 6 (limit: 5019)
    Memory: 4.2M
      CPU: 0.000 CPU(s) @ 2.60GHz
     CGroup: /system.slice/zabbix-agent.service
             └─26645 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
                  ├─26646 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
                  ├─26647 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                  ├─26648 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                  ├─26649 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                  ├─26650 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

dic 06 15:29:26 localhost.localdomain systemd[1]: Starting Zabbix Agent...
dic 06 15:29:26 localhost.localdomain systemd[1]: zabbix-agent.service: Can't open PID file /run/zabbix-agent.pid (yet): No such file or directory
dic 06 15:29:26 localhost.localdomain systemd[1]: Started Zabbix Agent.
```

FrontEnd

13. Configuramos zabbix desde un navegador desde otro host accediendo con la IP del servidor, en mi caso **192.168.56.105/zabbix**

Cuando accedemos a esa página web nos aparecerá una bienvenida y la versión que está funcionando:



14. Chequeamos de que lo que hemos instalado funciona y de que está bien configurado

	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](#)

15. Conectamos con la base de datos MySQL

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

Database type	<input type="button" value="MySQL"/>
Database host	<input type="text" value="localhost"/>
Database port	<input type="text" value="0"/> 0 - use default port
Database name	<input type="text" value="zabbix"/>
User	<input type="text" value="zabbix"/>
Password	<input type="password" value="*****"/>

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

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



GANA UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



16. Vemos si los datos están bien antes de terminar la instalación, puesto que si avanzamos no podremos deshacer el cambio

No seguro | 192.168.56.105/zabbix/setup.php

ZABBIX

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	zabbix-ubuntu

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

17. Una vez terminado nos mandará a la zona de administración, donde habrá que iniciar sesión como administrador <http://192.168.56.105/zabbix/index.php>, por defecto la instalación pone un usuario **Admin** y una contraseña **zabbix**

192.168.56.105/zabbix/index.php

ZABBIX

Username

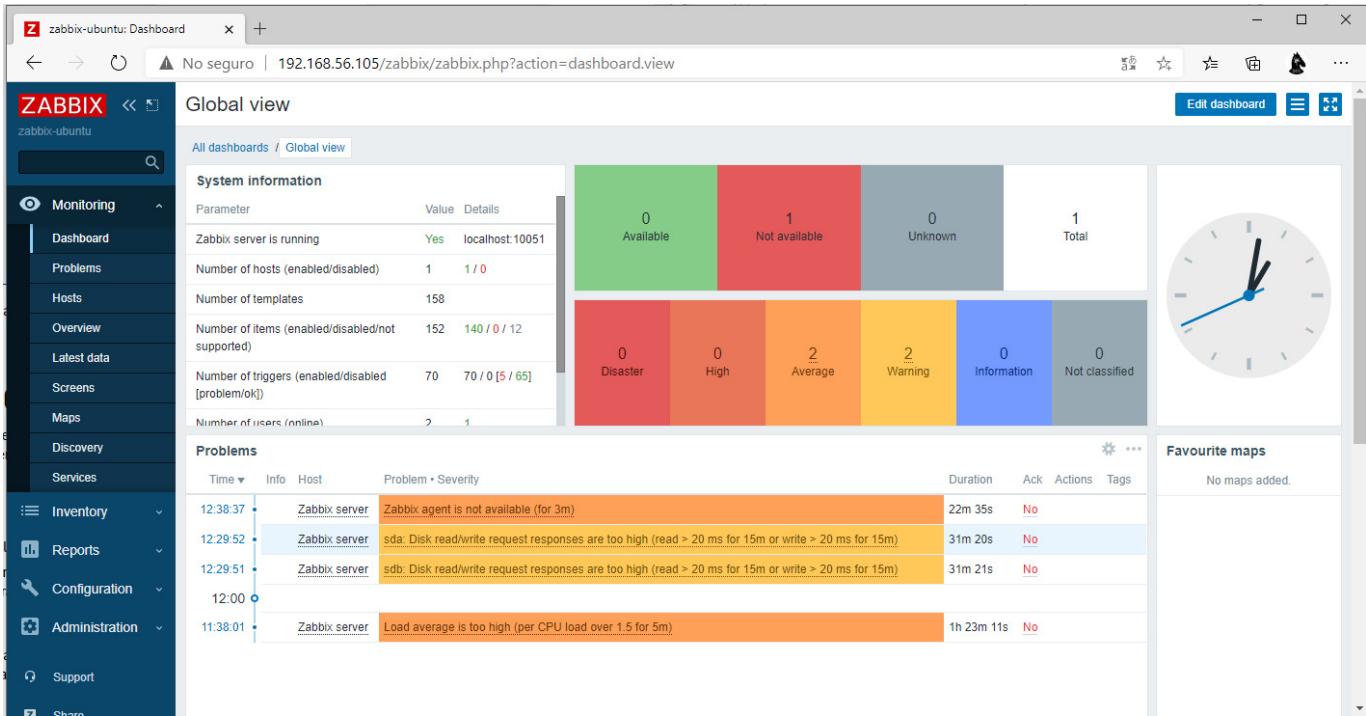
Password

Remember me for 30 days

Sign in

Help • Support

18. Una vez dentro podemos ver la monitorización del servidor de Ubuntu



Configurar la monitorización de los servicios SSH y HTTP

Creación del host en Ubuntu Server

- Para ello debemos crear un host, donde el cual habilitaremos los servicios SSH y HTTP, esto se encuentra en **configuration/host/creáte host**

The screenshot shows the Zabbix Hosts configuration page. The left sidebar highlights the "Configuration" section, specifically the "Hosts" item. The main form allows creating a new host with fields for Host groups, Templates, Name, DNS, IP, Port, Monitored by (Any, Server, Proxy), and Tags. A table lists existing hosts, including "Zabbix server" with details like IP 127.0.0.1:10050 and status Enabled. The "Create host" button is highlighted with a red box.

RUBEN GIRELA CASTELLON

12

WUOLAH

Una vez dentro, rellenamos los siguientes campos:

Hosts

* Host name: Ubuntu Server
 Visible name: Ubuntu Server
 * Groups: Linux servers, Virtual machines, Zabbix servers
 * Interfaces: Type: Agent, IP address: 192.168.56.105, DNS name: (empty), Connect to: IP, Port: 10051, Default: checked
 Description: (empty)
 Monitored by proxy: (no proxy)
 Enabled: checked
 Add, Cancel

- **Nombre del Host:** Ubuntu Server
- **Nombre visible (externamente):** Ubuntu Server
- **Grupos:**
 - Linux Servers
 - Virtual machine
 - Zabbix servers
- **IP del agente:** 192.168.56.105
- **Puerto:** 10051

Esos son los campos que he llenado, para la monitorización, que creo que son necesarios, el resto son opcionales. Por defecto, viene habilitado para que se active, con lo cual no será necesario activarlo, porque se activa solo.

	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption
<input type="checkbox"/>	Ubuntu Server	Applications	Items	Triggers	Graphs	Discovery	Web	192.168.56.105: 10051			Enabled	ZBX SNMP JMX IPMI	NONE

Creación de los templates para Ubuntu Server

Esto nos sirve para poder monitorizar los servicios del servidor, en nuestro caso SSH y HTTP

Para ello modificamos el host creado, y nos vamos a la pestaña de **templates**, una vez dentro añadimos los siguientes templates:

- **Template OS Linux by Zabbix agent** (indicamos el agente)
- **Template App SSH Service** (para monitorizar el SSH)
- **Template App Apache by HTTP** (para monitorizar el HTTP)
- **Template App Zabbix Server** (indicamos el servidor)

De tal forma que se me queda así, nota los templates van organizados por grupos (SO, APPS, NETWORK, ...), haciendo que sea más intuitivo a la hora de buscar:

Linked templates	Name	Action
	Template App Apache by HTTP	Unlink Unlink and clear
	Template App SSH Service	Unlink Unlink and clear
	Template App Zabbix Server	Unlink Unlink and clear
	Template OS Linux by Zabbix agent	Unlink Unlink and clear

Pero, tenemos un problema, y es que el puerto SSH lo habíamos cambiado en la práctica 2, que por defecto era 22, por 22022, para eso vamos a modificar el template SSH que por defecto trae el 22.

Para ello vamos a **templates**, seleccionamos el **template App SSH Service**, y nos vamos al apartado **ítems** y seleccionamos **SSH service is running**, dentro cambiamos únicamente el campo **Key**, donde tenemos que especificar el puerto SSH que utiliza **net.tcp.service[ssh,,22022]**, de tal forma que el tercer campo es el puerto, por eso esta así dentro de []. Una vez realizado le damos a **update**, para actualizar los cambios hechos.



GANA UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



The screenshot shows the Zabbix interface for creating a new item. The 'Name' field is set to 'SSH service is running'. The 'Type' is selected as 'Simple check'. The 'Key' field contains 'net.tcp.service[ssh,,22022]'. Other fields like 'User name' and 'Password' are empty. The 'Type of information' is set to 'Numeric (unsigned)'. The 'Update interval' is '1m'. A custom interval is defined with a flexible type, an interval of '50s', and a period from '1-7:00:00-24:00'. The 'Action' button is visible.

Creación del host para CentOS

- Para ello debemos crear un host, donde el cual habilitaremos los servicios SSH y HTTP, esto se encuentra en **configuration/host/creáte host**. Una vez dentro, rellenamos los siguientes campos:
 - Nombre del Host:** centOS
 - Nombre visible (externamente):** centOS
 - Grupos:**
 - Discovered hosts (para diferenciar entre Ubuntu server y el servidor de centOS)
 - IP del agente:** 192.168.56.110 (para que se conecte a centOS)
 - Puerto:** 10050

Hosts

The screenshot shows the 'Hosts' configuration page. A new host is being created with the following details:

- Host name:** centOS
- Visible name:** (empty)
- Groups:** Zabbix servers
- Interfaces:** Agent IP address: 192.168.56.110, Connect to: IP, Port: 10050, Default checked
- Description:** (empty)
- Monitored by proxy:** (no proxy)
- Enabled:** checked

 Buttons at the bottom include Update, Clone, Full clone, Delete, and Cancel.

Esos son los campos que he rellenado, para la monitorización, que creo que son necesarios, el resto son opcionales. Por defecto, viene habilitado para que se active, con lo cual no será necesario activarlo, porque se activa solo.

Resultado

Name ▾	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
centOS	Applications	Items	Triggers	Graphs	Discovery	Web	192.168.56.110: 10050			Enabled	ZBX SNMP JMX IPMI	NONE		

Añadir los templates para CentOS

Esto nos sirve para poder monitorizar los servicios que nos interesa, pero en este caso para el host de CentOS.

Para ello modificamos el host creado, y nos vamos a la pestaña de **templates**, una vez dentro añadimos los siguientes templates:

- **Template OS Linux by Zabbix agent** (indicamos el agente)
- **Template App SSH Service** (para monitorizar el SSH)
- **Template App Apache by HTTP** (para monitorizar el HTTP)

De tal forma que se me queda asi, nota los templates van organizados por grupos (SO, APPS, NETWORK, ...), haciendo que sea más intuitivo a la hora de buscar:

Hosts

Linked templates		Action
Name	Template App Apache by HTTP	Unlink Unlink and clear
Name	Template App SSH Service	Unlink Unlink and clear
Name	Template Module Zabbix agent	Unlink Unlink and clear

Link new templates

type here to search Select

Buttons: Update, Clone, Full clone, Delete, Cancel

Y no tenemos que modificar el template ssh, porque ya se ha modificado anteriormente de forma general, de tal forma que en todos los que usen ese template tendrá el puerto 22022.

Monitorización de los servicios SSH y HTTP

Monitorización en Ubuntu Server

Para ver la monitorización, nos vamos a **monitoring/dashboard** y hay dentro, en mi caso he añadido otro espacio de monitorización, donde creo varios **widgets**, con el host creado y templates seleccionados, ejemplo de añadir:

Add widget

Type Plain text Show header

Name default

Refresh interval Default (1 minute)

* Items

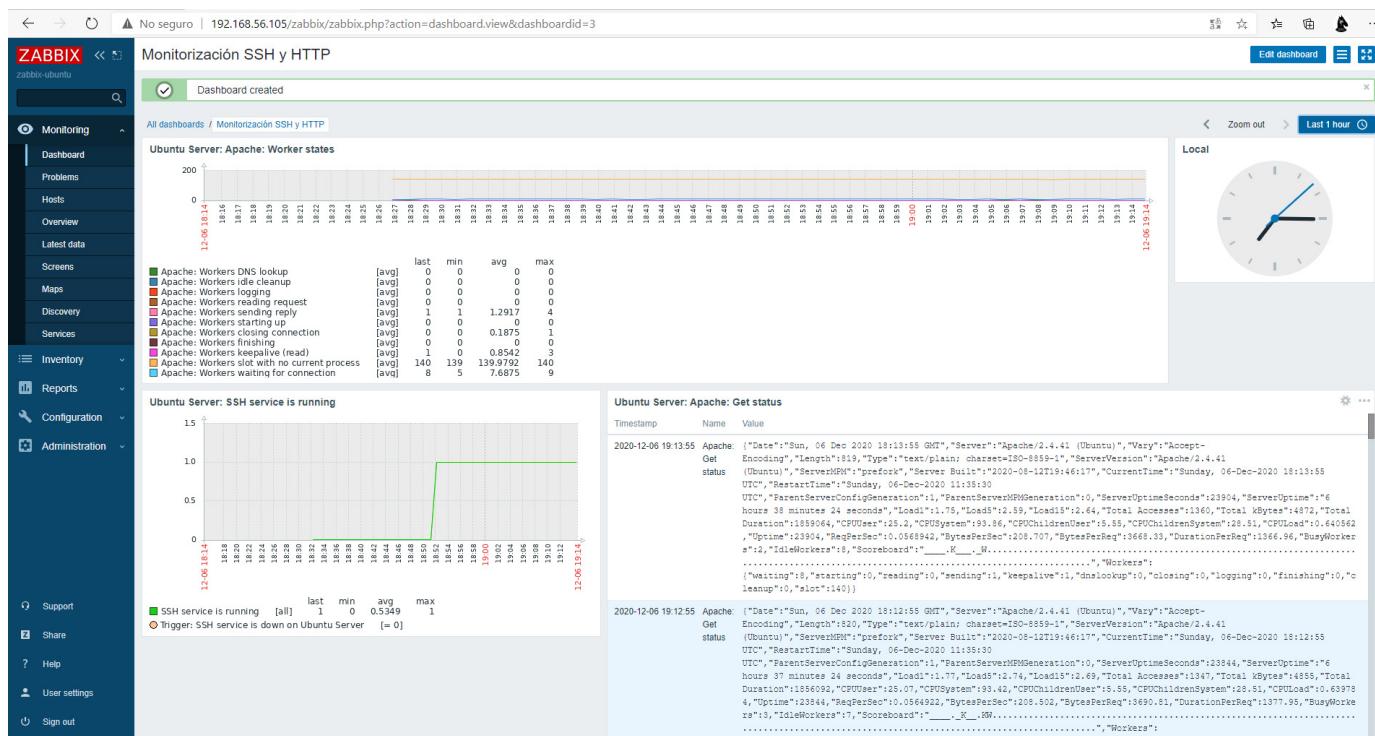
Items location Left Top

* Show lines

Show text as HTML

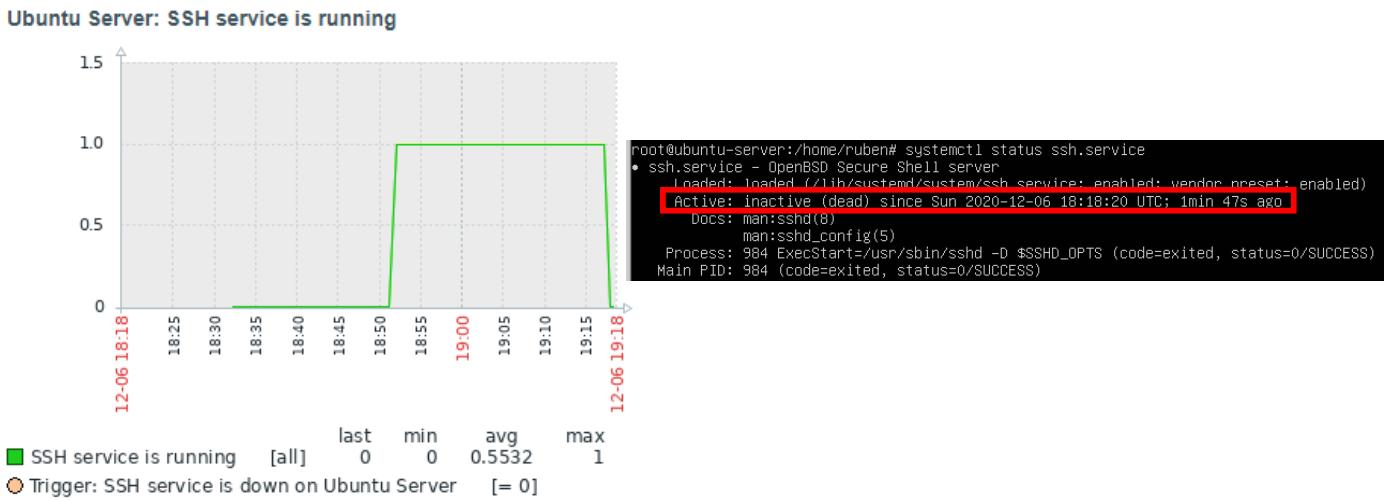
Dynamic items

Resultado final:



De tal forma que se puede ver la monitorización de forma gráfica del HTTP y debajo de forma gráfica el servicio SSH y en modo texto información del estado del servicio HTTP.

Para ver si monitoriza bien, apagamos el servicio SSH en Ubuntu server, de tal forma que si monitoriza bien:



Monitorización en CentOS

Hacemos lo mismo, pero para CentOS, pero no nos funcionara, ya que el item SSH y HTTP esta configurado con el del Ubuntu Server, para ello debemos crear 2 items nuevos, accedemos al host creado en CentOS, **ítems, créate ítem**

En el SSH es igual que el de Ubuntu server, salvo que le indicamos la IP del host CentOS, en los campos Key y Host interface, el resto de campos se deja igual.

Items

All hosts / CentOS Enabled ZBX SNMP JMX IPMI Applications 13 Items 65 Triggers 20 Graphs 11 Discovery rules 4 Web scenarios

Item Preprocessing

* Name	SSH service
Type	Zabbix agent
* Key	net.tcp.service[ssh,192.168.56.110 22022]
* Host interface	192.168.56.110 : 10050
Type of information	Numeric (unsigned)
Units	
* Update interval	1m
Custom intervals	
Type	Flexible
Interval	50s
Period	1-7:00:00-24:00
Action	Remove
Add	
* History storage period	Do not keep history
Storage period	90d
* Trend storage period	Do not keep trends
Storage period	365d
Show value	As is
show value mappings	



GANAR UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



Comprobamos que funciona antes de añadirlo, y vemos que si funciona, puesto que nos devuelve un 1, de que esta activo el servicio

Test item

Get value from host

Host address: 192.168.56.110 Port: 10050

Proxy: (no proxy)

Value: 1 Time: now

Previous value: Prev. time:

End of line sequence: LF CRLF

Result: Result converted to Numeric (unsigned) **1**

Get value and test **Cancel**

Enabled

Y en el HTTP es igual al del SSH, salvo que en el campo **Key** cambiamos ssh por http

Items

All hosts / centOS Enabled ZBX SNMP JMX IPMI Applications 13 Items 66 Triggers 20 Graphs 11 Discovery rules 4 Web scenarios

Item Preprocessing

* Name: HTTP service
Type: Zabbix agent
* Key: net.tcp.service[http,192.168.56.110.80]
* Host interface: 192.168.56.110 : 10050
Type of information: Numeric (unsigned)
Units:
* Update interval: 1m

Hacemos un test de que funciona y efectivamente funciona, puesto que nos devuelve un 1

Test item

Get value from host

Host address: 192.168.56.110 Port: 10050

Proxy: (no proxy)

Value: 1 Time: now

Previous value: Prev. time:

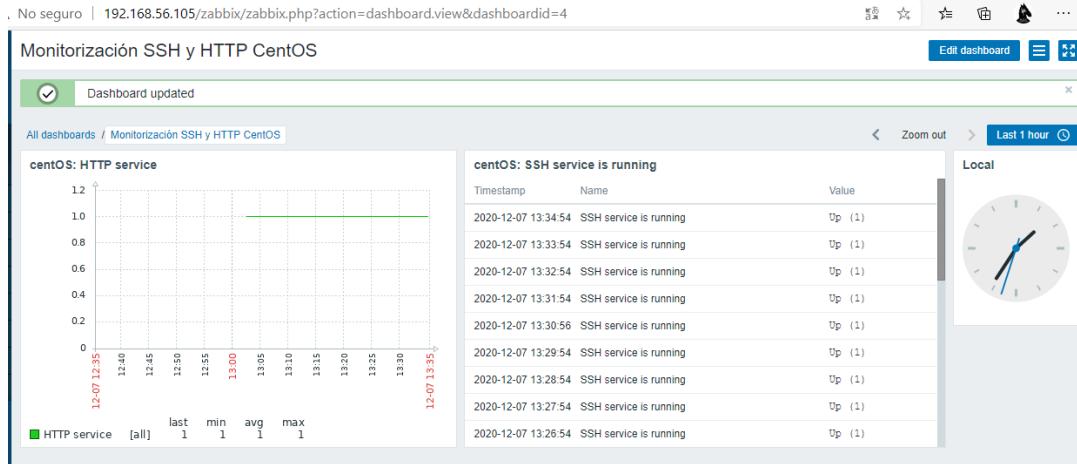
End of line sequence: LF CRLF

Result: Result converted to Numeric (unsigned) **1**

Get value and test **Cancel**

Enabled





y hacemos una prueba de que funciona, paramos el servicio ssh:

```
[root@localhost ruben]# systemctl stop sshd
[root@localhost ruben]# systemctl status sshd
● sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Sun 2020-12-06 16:22:45 EST; 7s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
   Process: 846 ExecStart=/usr/sbin/sshd -D $OPTIONS ${CRYPTO_POLICY} (code=exited, status=0/SUCCESS)
 Main PID: 846 (code=exited, status=0/SUCCESS)

dic 06 13:52:05 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
dic 06 13:52:06 localhost.localdomain sshd[846]: Server listening on 0.0.0.0 port 22022.
dic 06 13:52:06 localhost.localdomain sshd[846]: Server listening on :: port 22022.
dic 06 13:52:06 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
dic 06 16:22:45 localhost.localdomain systemd[1]: Stopping OpenSSH server daemon...
dic 06 16:22:45 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
[root@localhost ruben]#
```

centOS: SSH service is running		
Timestamp	Name	Value
2020-12-07 11:33:54	SSH service is running	Down (0)
2020-12-07 11:32:54	SSH service is running	Down (0)
2020-12-07 11:31:54	SSH service is running	Up (1)
2020-12-07 11:30:54	SSH service is running	Up (1)
2020-12-07 11:29:54	SSH service is running	Up (1)

Instalación y configuración Ansible en Ubuntu Server

Para instalarlo, primero tenemos que actualizar los repositorios con **apt update**, una vez hecho eso instalamos y añadimos los repositorios de ansible

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

Comprobamos que se ha instalado correctamente con el comando `ansible --version`

```
root@ubuntu-server:/home/ruben# ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jul 28 2020, 12:59:40) [GCC 9.3.0]
root@ubuntu-server:/home/ruben#
```

A continuación, vamos a la lista de ordenadores con los que trabajara, o dicho de otro modo a las maquinas a las que vamos a conectar con ansible, esta se encuentra en `/etc/ansible/hosts`, donde dentro añadiremos el nombre del modulo con el que llamaremos en ansible, el puerto, puesto que no usa el por defecto puerto 22, la IP de la maquina virtual CentOS a la que se conectara, el usuario con el que se conectara a CentOS (como hemos bloqueado el usuario root solo puedo acceder con el usuario normal) y parámetros del ssh, en el que le indicamos que se conecte por ssh.

```
#192.168.100.10
centos ansible_port=22022 ansible_host=192.168.56.110 ansible_user=ruben ansible_ssh_common_args=' -o StrictHostKeyChecking=no'

# 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" 45L, 1109C
```

17,53 Top

Guardamos los cambios y también tenemos que modificar el archivo de configuración de ansible que se encuentra en `/etc/ansible/ansible.cfg`, donde descomentaremos el campo:

- **Inventory = /etc/ansible/hosts** (donde accederá a las maquinas remotas)
- **Remote_port = 22022** (indicamos el puerto remoto)
- **ssh_args = ...** (con esto indicamos los parámetros ssh para conectarme a CentOS)

```
[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
```

```
[ssh_connection]
# ssh arguments to use
# Leaving off ControlPersist will result in poor performance, so use
# paramiko on older platforms rather than removing it, -C controls compression use
ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
```

Pero esto así no funcionara, puesto que nos falta comunicarnos con la maquina de CentOS para que funcione ansible, para ello tenemos que genera las claves pública y privada en nuestro Ubuntu Server y le mandamos la clave publica al servidor de CentOS, para ello uso 2 comandos:

- **ssh-keygen** (para generar las claves)
- **ssh-copy-id <ip> -p <port>** (para enviar la clave publica a la maquina remota)

En mi caso lo genero en Ubuntu Server y se la envió a CentOS Server

```
ruben@ubuntu-server:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ruben/.ssh/id_rsa):
Created directory '/home/ruben/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ruben/.ssh/id_rsa
Your public key has been saved in /home/ruben/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:jRMp/HznAOIeEP7k4835gvanfUKZ858hEUheYnfmBdI ruben@ubuntu-server
The key's randomart image is:
+---[RSA 3072]----+
|   .     + +... |
| . o   = =E. |
| o = + o . . |
| * = = . |
| * S +oo |
| o = +=+ . |
| o+. oo . |
| o .oo o. o |
| . .o+oo .o |
```

```
+---[SHA256]----+
ruben@ubuntu-server:~$ ssh-copy-id 192.168.56.110 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ruben/.ssh/id_rsa.pub"
The authenticity of host '[192.168.56.110]:22022 ([192.168.56.110]:22022)' can't be established.
ECDSA key fingerprint is SHA256:t2knLCIMRRChx6RLN5oYu+UuizFiWt+c8jdx2RJhFGQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ruben@192.168.56.110's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '22022' '192.168.56.110'"
and check to make sure that only the key(s) you wanted were added.

ruben@ubuntu-server:~$
```

Comprobamos que se puede conectar por ssh a CentOS

```
ruben@ubuntu-server:~$ ssh 192.168.56.110 -p 22022
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Sun Dec  6 13:52:34 2020
[ruben@localhost ~]$ exit
logout
Connection to 192.168.56.110 closed.
ruben@ubuntu-server:~$ _
```



GANAR UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



Y por último probamos que desde ansible se puede conectar a ese servidor haciendo un ping, con **-m** le indicamos el modulo a conectar, en mi caso yo la he llamado centos, si quisieramos conectarnos a todas las maquinas haciendo ping basta con poner **ansible -m ping all**

```
ruben@ubuntu-server:~$ ansible -m ping centos
centos | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": false,
    "ping": "pong"
}
ruben@ubuntu-server:~$
```

Ya podemos conectarnos a CentOS por ansible, podemos mandarle comandos a centos desde Ubuntu usando el comando **ansible -m Shell -a '<comando>' all** (all es para todos los hosts)

```
ruben@ubuntu-server:~$ ansible -m shell -a 'hostname' all
centos | CHANGED | rc=0 >>
localhost.localdomain
ruben@ubuntu-server:~$
```

En mi caso le he puesto que muestre el nombre del host remoto, o que nos muestre el sistema de archivos:

```
ruben@ubuntu-server:~$ ansible -m shell -a 'df -h' all
centos | CHANGED | rc=0 >>
Filesystem      Size  Used Avail Use% Mounted on
/devtmpfs       393M   0   393M  0% /dev
tmpfs          410M   0   410M  0% /dev/shm
tmpfs          410M  5.7M  404M  2% /run
tmpfs          410M   0   410M  0% /sys/fs/cgroup
/dev/mapper/c1-root  6.2G  1.8G  4.5G  29% /
/dev/sda1     976M 130M  790M  15% /boot
/dev/mapper/raid1-newvar_crypt  2.9G 337M  2.4G  13% /var
tmpfs          82M   0   82M  0% /run/user/1000
ruben@ubuntu-server:~$
```

Como se puede ver podemos verificar que los comandos se lo mandan a CentOS, porque por defecto los grupos de volúmenes lo hace con **cl. CHANGED** nos indica que la maquina ha cambiado algo en su configuración, pero esto no nos afecta, siempre que no se cambie la configuración del sshd de CentOS.

Pero no solo mostrar contenido, sino también hacer cambios o añadir, pongo como ejemplo la creación de un nuevo usuario llamado **testuser**.

Los parámetros: **-a** indicamos que vamos a meter argumentos al módulo, **-K** para que nos pida contraseña y el **-b** para ejecutar operaciones con conversión.

```
ruben@ubuntu-server:~$ ansible -b -K -m user -a 'name=testuser' all
BECOME password:
centos | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/libexec/platform-python"
    },
    "changed": true,
    "comment": "",
    "create_home": true,
    "group": 1001,
    "home": "/home/testuser",
    "name": "testuser",
    "shell": "/bin/bash",
    "state": "present",
    "system": false,
    "uid": 1001
}
ruben@ubuntu-server:~$
```

Y podemos ver que se ha creado ese usuario en CentOS e incluso acceder como ese usuario.

```
[root@localhost ruben]# su testuser
[testuser@localhost ruben]$ ls -l
ls: no se puede abrir el directorio '.': Permission denied
[testuser@localhost ruben]$ cd /home/testuser/
[testuser@localhost ~]$ ls -l
total 0
[testuser@localhost ~]$ mkdir mi_directorio
[testuser@localhost ~]$ ls -l
total 0
[drwxrwxr-x. 2 testuser testuser 6 dic 6 22:21 mi_directorio
[testuser@localhost ~]$ _
```

Pero esto no nos facilita mucho hacer operaciones de forma remota a CentOS, puesto que, si necesitas hacer 100 operaciones diarias, esto no es viable. Para ello existen los **playbook**, que nos permite ejecutar comandos a máquinas remotas con ansible, para crear un playbook, tenemos que tener las siguientes carpetas **roles**, **basic** y **tasks** (si no están creadas se crean), donde hay crearemos nuestros tareas (**task**), en mi caso como no están creadas esas carpetas las creo:

```
root@ubuntu-server:/home/ruben# mkdir /etc/ansible/roles
root@ubuntu-server:/home/ruben# mkdir /etc/ansible/roles/basic
root@ubuntu-server:/home/ruben# mkdir /etc/ansible/roles/basic/tasks
root@ubuntu-server:/home/ruben# cd /etc/ansible/roles/basic/tasks
root@ubuntu-server:/etc/ansible/roles/basic/tasks#
```

Una vez dentro de la carpeta tasks creamos un archivo **main.yml**, que tendrá la tarea a realizar, donde lo que haremos es parar el servicio **httpd** de CentOS:

```
- name: Stop httpd
  service:
    name=httpd
    state=stopped
~
```

Donde le indico el nombre de la tarea, el servicio a modificar y el estado que tendrá.

Después nos vamos a la carpeta **ansible** y creamos nuestro **playbook.yml**, donde dentro especificamos el host al que se le va a ejecutar los comandos, **become** siempre a true para ejecutar los comandos como superusuario e indicamos el rol o roles que va a tener, en mi caso como solo hay 1 el **basic**, pero se puede tener varios roles:

```
---
- hosts: centos
  become: true
  roles:
    - basic
~
```

Y luego es tan simple como ejecutar el siguiente comando para ejecutar el playbook **ansible-playbook -K <archivo_playbook>**, como tenemos activado **become**, nos pedirá la contraseña de superusuario para aplicar el cambio.

Antes de ejecutar el comando comprobamos en centos de que el servicio httpd está arrancado

```

CentOS2_practica3 (instalación y configuración zabbix agent) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
    └─php-fpm.conf
    Active: active (running) since Sun 2020-12-06 23:34:05 EST; 8s ago
      Docs: man:httpd.service(8)
    Main PID: 36183 (httpd)
    Status: "Started, listening on: port 80"
      Tasks: 213 (limit: 5019)
     Memory: 33.0M
    CGroup: /system.slice/httpd.service
            ├─36183 /usr/sbin/httpd -DFOREGROUND
            ├─36184 /usr/sbin/httpd -DFOREGROUND
            ├─36185 /usr/sbin/httpd -DFOREGROUND
            ├─36186 /usr/sbin/httpd -DFOREGROUND
            ├─36188 /usr/sbin/httpd -DFOREGROUND

dic 06 23:34:05 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 06 23:34:05 localhost.localdomain httpd[36183]: AH00558: httpd: Could not reliably determine the fully qualified domain name for the local host
dic 06 23:34:05 localhost.localdomain httpd[36183]: Server configured, listening on: port 80
dic 06 23:34:05 localhost.localdomain systemd[1]: Started The Apache HTTP Server.

~
```

Y vemos que esta arrancado, si ejecutamos el **playbook** con Ansible:

```

CentOS2_practica3 (instalación y configuración zabbix agent) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
dic 06 23:34:05 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 06 23:34:05 localhost.localdomain httpd[36183]: AH00558: httpd: Could not reliably determine the fully qualified domain name for the local host
dic 06 23:34:05 localhost.localdomain httpd[36183]: Server configured, listening on: port 80
dic 06 23:34:05 localhost.localdomain systemd[1]: Started The Apache HTTP Server.

~
```



```

Ubuntu Server p3 (Instalación y configuración zabbix) [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
roles/basic/tasks/main.yml" 4L, 66C written
root@ubuntu-server:/etc/ansible# ansible-playbook -K playbook.yml
BECOME password:
PLAY [centos] *****
TASK [Gathering Facts] *****
ok: [centos]
TASK [basic : Stop httpd] *****
changed: [centos]
PLAY RECAP *****
centos : ok=2 changed=1 unreachable=0 failed=0 skipped=0 rescued=0 ignored=0
root@ubuntu-server:/etc/ansible#
```

Vemos que se ha detenido el servicio Apache HTTP.

Pero podemos hacer más, podemos incluso instalar servicios a CentOS

```
- name: Stop httpd
  service:
    name=httpd
    state=stopped
- name: Installing latest version GIT
  yum:
    name=git
    state=latest
```

Como en centos no esta instalado apt, usamos yum.

En este caso indicamos lo mismo el nombre a mostrar, **yum** para indicar que se va a usar el servicio yum, **name** el paquete a instalar y **state** el estado que queremos que haga con el servicio, en mi caso **latest**, porque ya estaba instalado de antes, pero para instalar un servicio que no esta se usaría **installed**. De tal forma que en este ejemplo nos instalara la última versión de GIT en CentOS si la hay.

```
"roles/basic/tasks/main.yml" 8L, 143C written
root@ubuntu-server:/etc/ansible# ansible-playbook -K playbook.yml
BECOME password:

PLAY [centos] *****
TASK [Gathering Facts] *****
ok: [centos]

TASK [basic : Stop httpd] *****
ok: [centos]

TASK [basic : Installing latest version GIT] *****
changed: [centos]

PLAY RECAP *****
centos : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
root@ubuntu-server:/etc/ansible#
```

Vemos que se ha hecho un cambio, y es que ha actualizado GIT a la ultima versión. Si no ha habido cambio es que no ha hecho nada en CentOS.

Si tuviese instalado **apt** la instalación de paquetes seria de esta forma:

```
- name: Installing GIT
  apt:
    pkg=git state=installed
  ~
```

Es igual que en yum, pero en vez de **name** es **pkg**.

Incluso si queremos instalar varios paquetes podemos hacer en un único comando la instalación y una lista de paquetes a instalar. Ejemplo:

```
- name: Installing latest version GIT, VIM and AT
  yum:
    name={{ item }}
    state=latest
  with_items:
    - git
    - vim
    - at
```



GANAR UN FIAT 500 CON BURN ENERGY

1 LATA = 1 PARTICIPACIÓN



```
"roles/basic/tasks/main.yml" 12L, 198C written
root@ubuntu-server:/etc/ansible# ansible-playbook -K playbook.yml
BECOME password:

PLAY [centos] *****
TASK [Gathering Facts] *****
ok: [centos]

TASK [basic : Stop httpd] *****
ok: [centos]

TASK [basic : Installing latest version GIT, VIM and AT] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}`",
please use `name: ['git', 'vim', 'at']` and remove the loop. This feature will be removed in
version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
changed: [centos] => (item=['git', 'vim', 'at'])

PLAY RECAP *****
centos : ok=3    changed=1    unreachable=0    failed=0    skipped=0    rescued=0
          ignored=0

root@ubuntu-server:/etc/ansible#
```

Aunque nos recomienda usar un bucle en vez de hacerlo así, pero efectúa correctamente los cambios.

También podemos transferir archivos desde nuestro Ubuntu Server a CentOS:

```
- name: Add file
  copy:
    src: /home/ruben/mi_archivo.txt
    dest: /home/ruben
    owner: root
    mode: '0644'
```

Indicamos que queremos copiar archivos desde nuestra máquina local a la remota con **copy**, **src** ruta del archivo en la máquina local, **dest** ruta de la máquina remota donde se almacenará el archivo, el usuario que tiene acceso al archivo, en mi caso solo root y los permisos que se le dará a ese archivo.

Creamos el archivo en la máquina local:

```
"roles/basic/tasks/main.yml" 15L, 299C written
root@ubuntu-server:/etc/ansible# ls -l /home/ruben/
total 16
-rw-r--r-- 1 root root 42 Dec  8 13:11 mi_archivo.txt
drwxr-xr-x 3 root root 4096 Nov  9 15:49 prueba_git
-rw-r--r-- 1 root root 4244 May 28 2020 zabbix-release_5.0-1+focal_all.deb
root@ubuntu-server:/etc/ansible#
```

Y vemos antes de ejecutar el comando ansible que en la máquina remota no está:

```
[ruben@localhost ~]$ ls -l /home/ruben/
total 0
[ruben@localhost ~]$ _
```

Ejecutamos el comando ansible con el playbook

```
"roles/basic/tasks/main.yml" 18L, 318C written
root@ubuntu-server:/etc/ansible# ansible-playbook -K playbook.yml
BECOME password:

PLAY [centos] *****
TASK [Gathering Facts] *****
ok: [centos]

TASK [basic : Stop httpd] *****
ok: [centos]

TASK [basic : Installing latest version GIT, VIM and AT] *****
[DEPRECATION WARNING]: Invoking "yum" only once while using a loop via squash_actions is
deprecated. Instead of using a loop to supply multiple items and specifying `name: "{{ item }}``,
please use `name: ['git', 'vim', 'at']` and remove the loop. This feature will be removed in
version 2.11. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
ok: [centos] => (item=['git', 'vim', 'at'])

TASK [basic : Add file] *****
changed: [centos]

PLAY RECAP *****
centos : ok=4    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
root@ubuntu-server:/etc/ansible#
```

Y vemos que se ha hecho un cambio, nos vamos a la maquina de CentOS y efectivamente se ha guardado el archivo:

```
[ruben@localhost ~]$ ls -l /home/ruben/
total 4
-rw-r--r--. 1 root root 42 dic  7 01:08 mi_archivo.txt
[ruben@localhost ~]$ _
```

Pero el usuario ruben, no puede ver el contenido ni modificarlo



Pero si accedemos como root, si podremos ver el contenido y modificarlo:

```
holaaa me he copiado de ubuntu a centooos
[root@localhost ruben]# vi /home/ruben/mi_archivo.txt
```

Aunque hay muchas más funciones como:

- mover archivos en la maquina remota

```
- name: Estado del archivo mi_archivo.txt
  stat: path=/home/ruben/mi_archivo.txt
  register: mi_archivo
- name: Move file
  command: mv /home/ruben/mi_archivo.txt /home/testuser
  when: mi_archivo.stat.exists
```

El primero comprueba el estado de ese archivo y el segundo mueve el archivo si existe.

```
TASK [basic : Move file] *****
changed: [centos]
PLAY RECAP *****
centos          : ok=6    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
root@ubuntu-server:/etc/ansible#
```

```
[root@localhost ruben]# ls -l /home/ruben/
total 0
[root@localhost ruben]# ls -l /home/testuser/
total 4
-rw-r--r--. 1 root      root      42 dic  7 01:17 mi_archivo.txt
drwxrwxr-x. 2 testuser  testuser   6 dic  6 22:21 mi_directorio
[root@localhost ruben]#
```

- y muchas más funcionalidades

Bibliografía

Zabbix

https://www.zabbix.com/download?zabbix=5.0&os_distribution=ubuntu&os_version=20.04_focal&db=mysql&ws=apache

<https://www.zabbix.com/documentation/5.0/manual/config/templates/template>

https://www.zabbix.com/documentation/4.0/manual/installation/install#installing_frontend

<https://www.zabbix.com/documentation/4.0/manual/quickstart/login>

<https://www.zabbix.com/documentation/4.0/manual/quickstart/host>

<https://www.zabbix.com/documentation/4.0/manual/quickstart/item>

<https://www.zabbix.com/documentation/2.2/manual/config/macros/usermacros>

<https://blog.zabbix.com/apache-http-server-monitoring/6972/>

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-centos-7>

Ansible

https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html#installing-ansible-on-ubuntu

https://docs.ansible.com/ansible/latest/user_guide/playbooks_intro.html#playbooks-intro

https://docs.ansible.com/ansible/latest/community/other_tools_and_programs.html#other-tools-and-programs

<https://github.com/pearofducks/ansible-vim>

https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html#yaml-syntax

<https://www.youtube.com/watch?v=icR-df2Olm8&feature=youtu.be>

<https://www.youtube.com/watch?v=pRZA9ymZXn0>

<https://www.howtoforge.com/tutorial/setup-new-user-and-ssh-key-authentication-using-ansible/>

<https://www.edureka.co/community/79444/how-to-configure-htpd-web-server-using-ansible-playbook>

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/service_module.html

<http://howto.eyeoncomputers.com/linux/install-htpd-using-ansible/>

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/apt_module.html

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/yum_module.html

<https://linuxbuz.com/linuxhowto/ansible-apt-module>

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/copy_module.html

<https://stackoverflow.com/questions/24162996/how-to-move-rename-a-file-using-an-ansible-task-on-a-remote-system>