

INGENIERÍA DE SERVIDORES (21-22)  
Grado en Ingeniería Informática  
Universidad de Granada

Arturo Alonso Carbonero

## Práctica 3 : Monitorización, Automatización y "Profiling"

---



A N S I B L E

# ÍNDICE

## **1. Introducción.**

## **2. Instalación de Zabbix en Ubuntu Server.**

### **2.1. Obtención del repositorio para Zabbix.**

### **2.2. Instalación de los servicios de Zabbix.**

### **2.3. Configuración de la base de datos de Zabbix.**

## **3. Configuración de PHP para el frontend.**

## **4. Comprobación del estado de los servicios.**

## **5. Habilitación del puerto de escucha de Zabbix.**

## **6. Configuración del agente de Zabbix.**

### **6.1. Configuración del agente en Ubuntu Server.**

### **6.2. Instalación y configuración del agente en CentOS.**

## **7. Zabbix.**

### **7.1. Configuración y acceso a zabbix desde el navegador.**

### **7.2. Configuración de la monitorización.**

#### **7.2.1. Creación del host de Ubuntu Server.**

#### **7.2.2. Creación del host de CentOS.**

## **8. Monitorización en Zabbix.**

### **8.1. Ubuntu Server.**

### **8.2. CentOS.**

## **9. Ansible.**

### **9.1. Instalación y configuración de Ansible en Ubuntu Server.**

### **9.2. Ejemplos de uso de Ansible.**

## **10. Referencias**

## 1. Introducción

Monitorización de los servicios SSH y HTTP (entre otros) desde una máquina con Ubuntu Server hacia sí misma y hacia una máquina con CentOS en Virtual Box.

Para la realización de la práctica, partiré de las máquinas utilizadas en la práctica 2, esto es, con una pila *Lamp* instalada (Linux + Apache + Mysql + PHP) y el servicio SSH pero sin fail2ban. Además, la instalación de las mismas es la que ofrecen por defecto, es decir, sin particiones como las de la práctica 1.

Las direcciones IP de las máquinas son las utilizadas a lo largo del curso:

**CentOs → 192.168.56.110**

**Ubuntu Server → 192.168.56.105**

El usuario que utilizaré será el empleado en las sesiones de prácticas anteriores y se construye de la siguiente forma: Dos primeras letras del segundo apellido + Dos primeras letras del primer apellido + Primera letra del nombre:

**Arturo Alonso Carbonero → **alcaa**.**

Las imágenes que aparecen en esta memoria contarán con una explicación de lo que se muestra justo debajo de las mismas.

## 2. Instalación de Zabbix en Ubuntu Server

### 2.1. Obtención del repositorio para Zabbix.

Para poder comenzar a trabajar con Zabbix es necesario instalar los servicios que ofrece. Para ello, es preciso obtener el repositorio con los paquetes necesarios mediante el comando siguiente:

```
alcaa@alcaa:~$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
--2021-12-05 18:44:38-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4244 (4.1K) [application/octet-stream]
Saving to: 'zabbix-release_5.0-1+focal_all.deb'

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

2021-12-05 18:44:39 (35.8 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]

alcaa@alcaa:~$
```

*Obtención del repositorio para Zabbix*

A continuación, se debe desempaquetar el contenido del repositorio con la opción **-i** para instalar el paquete en el directorio de paquetes:

```
alcaa@alcaa:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
[sudo] password for alcaa:
Selecting previously unselected package zabbix-release.
(Reading database ... 72024 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) ...
alcaa@alcaa:~$
```

### 2.2. Instalación de los servicios de Zabbix.

Una vez obtenido el contenido del repositorio, actualizamos los paquetes y procedemos con la instalación de los servicios de Zabbix. Instalaremos concretamente **zabbix-server-mysql** para la base de datos, **zabbix-frontend-php** para el frontend, **zabbix-agent** para poder hacer uso del agente de zabbix y **zabbix-apache-conf** para realizar ciertas modificaciones que se indicarán más adelante.

```
alcaa@alcaa:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4958 B]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1193 B]
Get:6 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:7 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [3232 B]
Fetched 345 kB in 1s (276 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
135 packages can be upgraded. Run 'apt list --upgradable' to see them.
alcaa@alcaa:~$
```

*Actualización de paquetes*

```
alcaa@alcaa:~$ sudo apt-get install zabbix-server-mysql zabbix-frontend-php zabbix-agent_
```

```
alcaa@alcaa:~$ sudo apt install zabbix-apache-conf
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  zabbix-apache-conf
0 upgraded, 1 newly installed, 0 to remove and 135 not upgraded.
Need to get 5900 B of archives.
After this operation, 18.4 kB of additional disk space will be used.
Get:1 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 zabbix-apache-conf all 1:5.0.18-1+fo
cal [5900 B]
Fetched 5900 B in 1s (8843 B/s)
Selecting previously unselected package zabbix-apache-conf.
(Reading database ... 74580 files and directories currently installed.)
Preparing to unpack .../zabbix-apache-conf_1%3a5.0.18-1+focal_all.deb ...
Unpacking zabbix-apache-conf (1:5.0.18-1+focal) ...
Setting up zabbix-apache-conf (1:5.0.18-1+focal) ...
Enabling conf zabbix.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

```
alcaa@alcaa:~$ sudo systemctl reload apache2
```

*Instalación de los servicios de Zabbix*

### 2.3. Configuración de la base de datos de Zabbix.

Es necesario configurar la base de datos que utilizará Zabbix. Para ello, crearemos la propia base de datos y un usuario al que otorgaremos todos los privilegios sobre la misma.

```
alcaa@alcaa:~$ sudo mysql -u root -p
```

```
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

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

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

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

mysql> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected, 2 warnings (0.02 sec)

mysql> create user zabbix@localhost identified by 'practicas,ISE';
Query OK, 0 rows affected (0.04 sec)

mysql> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.02 sec)

mysql> quit
Bye
alcaa@alcaa:~$ _
```

*Creación de la base de datos y del usuario*

Una vez creada la base de datos, obtenemos el esquema de datos de la misma:

```
alcaa@alcaa:~$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -u zabbix -p zabbix_
```

Finalmente, añadimos la contraseña de la base de datos al fichero de configuración del server de Zabbix situado en /etc: **/etc/zabbix/zabbix\_server.conf**

```
# 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=practicass.ISE

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

### Option: DBPort
# Database port when not using local socket.
#
# Mandatory: no
# Range: 1024-65535
# Default:
# DBPort=

-- INSERT --
```

124,25

12%

### 3. Configuración de PHP para el frontend

Configuramos el fichero **/etc/zabbix/apache.conf** donde debemos descomentar los parámetros correspondientes a la zona horaria e indicar la correcta (en este caso Madrid).

```
# Define /zabbix alias, this is the default
<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
-- INSERT --
```

11,1

Top

## 4. Comprobación del estado de los servicios

Comprobamos mediante el comando **systemctl status <servicio>** que los servicios están activos. Reiniciamos los servicios con el comando **sudo systemctl restart <servicio>** y, si no están activos, los activamos con **sudo systemctl start <servicio>** y, además, los habilitamos para que se activen automáticamente tras iniciar la máquina con el comando **sudo systemctl enable <servicio>**.

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

*Reinicio y habilitación de los servicios*

```
• zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-12-05 19:15:02 UTC; 6min ago
   Main PID: 18235 (zabbix_server)
     Tasks: 38 (limit: 1072)
    Memory: 29.7M
   CGroup: /system.slice/zabbix-server.service
           └─18235 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
             └─18236 /usr/sbin/zabbix_server: configuration syncer [syncd configuration in 0.05954s]
               └─18242 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.014s]
                 └─18243 /usr/sbin/zabbix_server: alerter #1 started
                   └─18244 /usr/sbin/zabbix_server: alerter #2 started
                     └─18245 /usr/sbin/zabbix_server: alerter #3 started
                       └─18246 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 0, processed 8 value]
                         └─18247 /usr/sbin/zabbix_server: preprocessing worker #1 started
                           └─18248 /usr/sbin/zabbix_server: preprocessing worker #2 started
                             └─18249 /usr/sbin/zabbix_server: preprocessing worker #3 started
                               └─18250 /usr/sbin/zabbix_server: lld manager #1 [processed 0 LLD rules, idle 5.990522s]
                                 └─18251 /usr/sbin/zabbix_server: lld worker #1 started
                                   └─18252 /usr/sbin/zabbix_server: lld worker #2 started
                                     └─18253 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes]
                                       └─18254 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.0>
                                         └─18255 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.002383 sec, idle 5>
                                           └─18256 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.001485 sec, idl>
                                             └─18257 /usr/sbin/zabbix_server: history syncer #1 [processed 0 values, 0 triggers in>
                                               └─18258 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0 triggers in>
                                                 └─18259 /usr/sbin/zabbix_server: history syncer #3 [processed 0 values, 0 triggers in>
                                                   └─18260 /usr/sbin/zabbix_server: history syncer #4 [processed 2 values, 2 triggers in>
                                                     └─18261 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.003278 sec>
                                                       └─18262 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.0>
                                                         └─18263 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000074 sec, idle>
                                                           └─18264 /usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.001132 sec, id>
                                                             └─18265 /usr/sbin/zabbix_server: poller #1 [got 1 values in 0.000198 sec, idle 1 sec]
                                                               └─18266 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000082 sec, idle 1 sec]
                                                                 └─18267 /usr/sbin/zabbix_server: poller #3 [got 0 values in 0.000069 sec, idle 1 sec]
                                                                   └─18275 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000038 sec, idle 1 sec]
```

*Comprobación del estado del server – `systemctl status zabbix-server`*

```
alcaa@alcaa:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-12-05 19:15:02 UTC; 4min 21s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 18225 (apache2)
    Tasks: 6 (limit: 1072)
   Memory: 20.5M
   CGroup: /system.slice/apache2.service
           └─18225 /usr/sbin/apache2 -k start
             └─18237 /usr/sbin/apache2 -k start
               └─18238 /usr/sbin/apache2 -k start
                 └─18239 /usr/sbin/apache2 -k start
                   └─18240 /usr/sbin/apache2 -k start
                     └─18241 /usr/sbin/apache2 -k start

Dec 05 19:15:01 alcaa systemd[1]: Starting The Apache HTTP Server...
Dec 05 19:15:02 alcaa apachectl[18224]: AH00558: apache2: Could not reliably determine the server's s
Dec 05 19:15:02 alcaa systemd[1]: Started The Apache HTTP Server.
```

### Comprobación del estado de Apache2

```
alcaa@alcaa:~$ sudo systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2021-12-05 19:15:02 UTC; 6min ago
     Main PID: 18227 (zabbix_agentd)
    Tasks: 6 (limit: 1072)
   Memory: 9.8M
   CGroup: /system.slice/zabbix-agent.service
           └─18227 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
             └─18228 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
               └─18229 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                 └─18230 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                   └─18231 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                     └─18232 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

Dec 05 19:15:01 alcaa systemd[1]: Starting Zabbix Agent...
Dec 05 19:15:02 alcaa systemd[1]: zabbix-agent.service: Can't open PID file /run/zabbix/zabbix_agen
Dec 05 19:15:02 alcaa systemd[1]: Started Zabbix Agent.
```

### Comprobación del estado del agente

## 5. Habilitación del puerto de escucha de Zabbix

Habilitamos el puerto de escucha que utiliza Zabbix, por defecto el 10051, en el firewall.

```
alcaa@alcaa:~$ sudo ufw allow 10051/tcp
Rule added
Rule added (v6)
alcaa@alcaa:~$ sudo ufw status
Status: active

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



## 6. Configuración del agente de Zabbix

### 6.1. Configuración del agente en Ubuntu Server.

El agente de Zabbix es necesario para que el servidor Zabbix pueda monitorizar las máquinas. Para poder hacer uso del agente en Ubuntu Server, es necesario añadir a los ficheros que se indican a continuación un parámetro que defina la IP del servidor Zabbix. En este caso, el server se encuentra en la máquina Ubuntu por lo que dicha dirección será **192.168.56.105**. El fichero a modificar es **/etc/zabbix/zabbix\_agentd.conf**

```
### 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
```

```
##### Active checks related

### 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
```

## 6.2. Instalación y configuración del agente en CentOS.

Obtenemos el repositorio necesario para instalar el agente y procedemos con la instalación del mismo:

```
[alcaa@localhost ~]$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
[sudo] password for alcaa:
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.xIZwG9: EncabezadoV4 RSA/SHA512 Signature, ID de clave a14fe591: NOKEY
Verifying...                               [100%]
Preparando...                               [100%]
Actualizando / instalando...
 1:zabbix-release-5.0-1.el8                [100%]
```

```
[alcaa@localhost ~]$ sudo yum install zabbix-agent
```

Una vez instalado en agente de Zabbix en la máquina con CentOS, habilitamos el puerto de escucha que utiliza el agente, que por defecto se trata del 10050, y recargamos el firewall para que se aplique la regla:

```
[alcaa@localhost ~]$ sudo firewall-cmd --add-port=10050/tcp --permanent
success
[alcaa@localhost ~]$ sudo firewall-cmd --reload
success
```

Añadimos los mismos parámetros que añadimos al fichero de configuración del agente en Ubuntu Server (En CentOS el fichero se encuentra en el mismo directorio y tiene el mismo nombre):

```
### 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=127.0.0.1:30051,zabbix.domain,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=
ServerActive=192.168.56.105
```

Para distinguir Ubuntu Server de CentOS, voy a cambiar el nombre del host en la máquina CentOS. Dicho nombre se ha de indicar en el parámetro que se muestra en la imagen siguiente en el mismo fichero de configuración del paso inmediatamente anterior:

```
### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the server.
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=
hostname=Zabbix-agent-CentOs
```

Es necesario habilitar las políticas de **SELinux**. En la práctica 2 se realizó dicha acción por lo que no será necesario repetirla. El comando para ello es el siguiente:

**sudo yum install policycoreutils-python-utils.noarch**

Comprobamos que el proceso se ha realizado con éxito mediante el comando **systemctl**, tal y como se llevó a cabo en Ubuntu Server.

```
lcalca@localhost ~]$ sudo systemctl restart zabbix-agent
lcalca@localhost ~]$ sudo 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 2021-12-05 14:44:16 EST; 8s ago
   Process: 28942 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
  Main PID: 28944 (zabbix_agentd)
    Tasks: 6 (limit: 5019)
   Memory: 3.9M
    CGroup: /system.slice/zabbix-agent.service
            └─28944 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
              └─28945 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
                └─28946 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                  └─28947 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                    └─28948 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                      └─28949 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

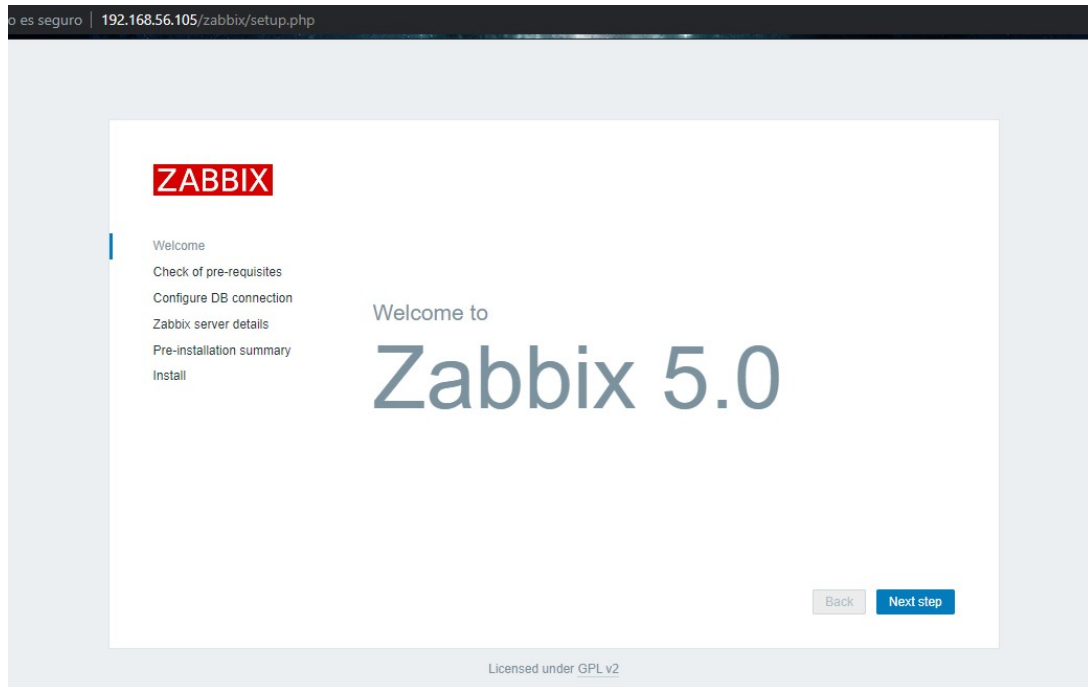
dic 05 14:44:16 localhost.localdomain systemd[1]: Starting Zabbix Agent...
dic 05 14:44:16 localhost.localdomain systemd[1]: zabbix-agent.service: Can't open PID file /run/za
dic 05 14:44:16 localhost.localdomain systemd[1]: Started Zabbix Agent.
```

*Agente de Zabbix activo en CentOS*

## 7. Zabbix

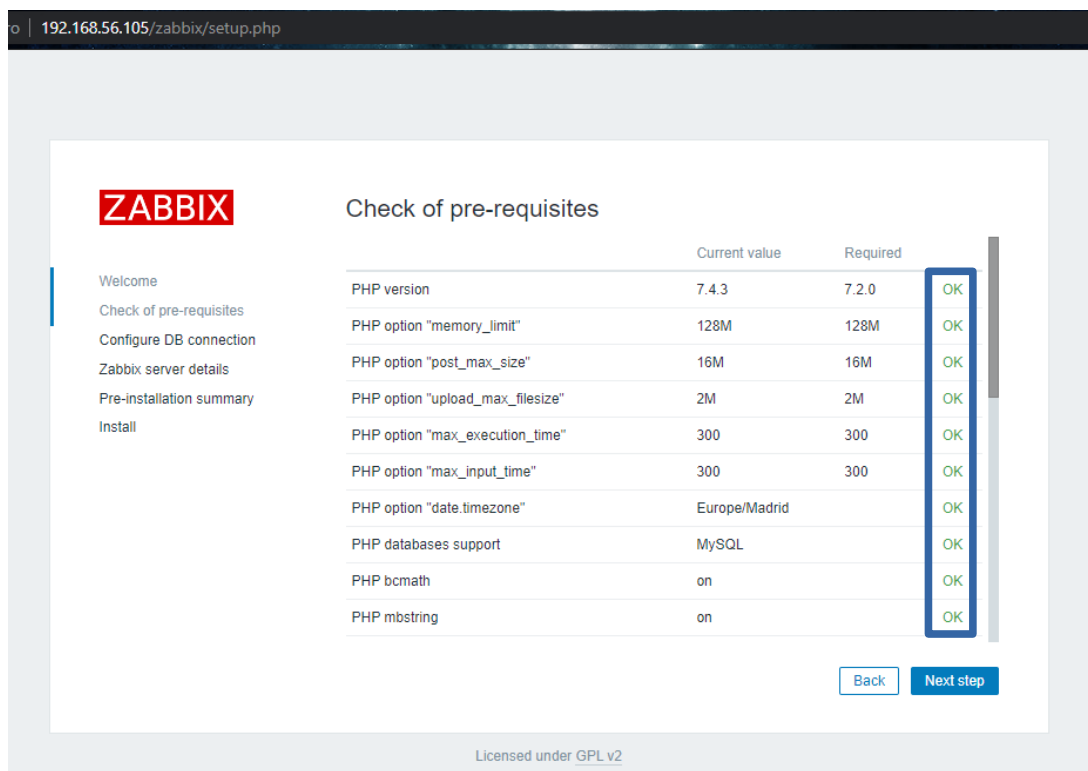
### 7.1. Configuración y acceso a zabbix desde el navegador.

Para comenzar a hacer uso de zabbix y monitorizar los servicios de las máquinas, es necesario acceder desde el navegador a la dirección **192.168.56.105/zabbix**.



*Pantalla de bienvenida, acceso correcto*

A continuación, debemos comprobar que los requisitos previos son correctos.



Realizamos la configuración para la conexión a la base de datos de Zabbix. Los datos introducidos deben ser adecuados ya que una vez indicados no podremos cambiarlos, por lo que debemos cerciorarnos de que son correctos.

guro | 192.168.56.105/zabbix/setup.php

**ZABBIX**

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

### Configure DB connection

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

Database type	MySQL	
Database host	localhost	
Database port	0	0 - use default port
Database name	zabbix	
User	zabbix	
Password	*****	

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

BackNext step

Licensed under [GPL v2](#)

guro | 192.168.56.105/zabbix/setup.php

**ZABBIX**

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

### Pre-installation summary

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

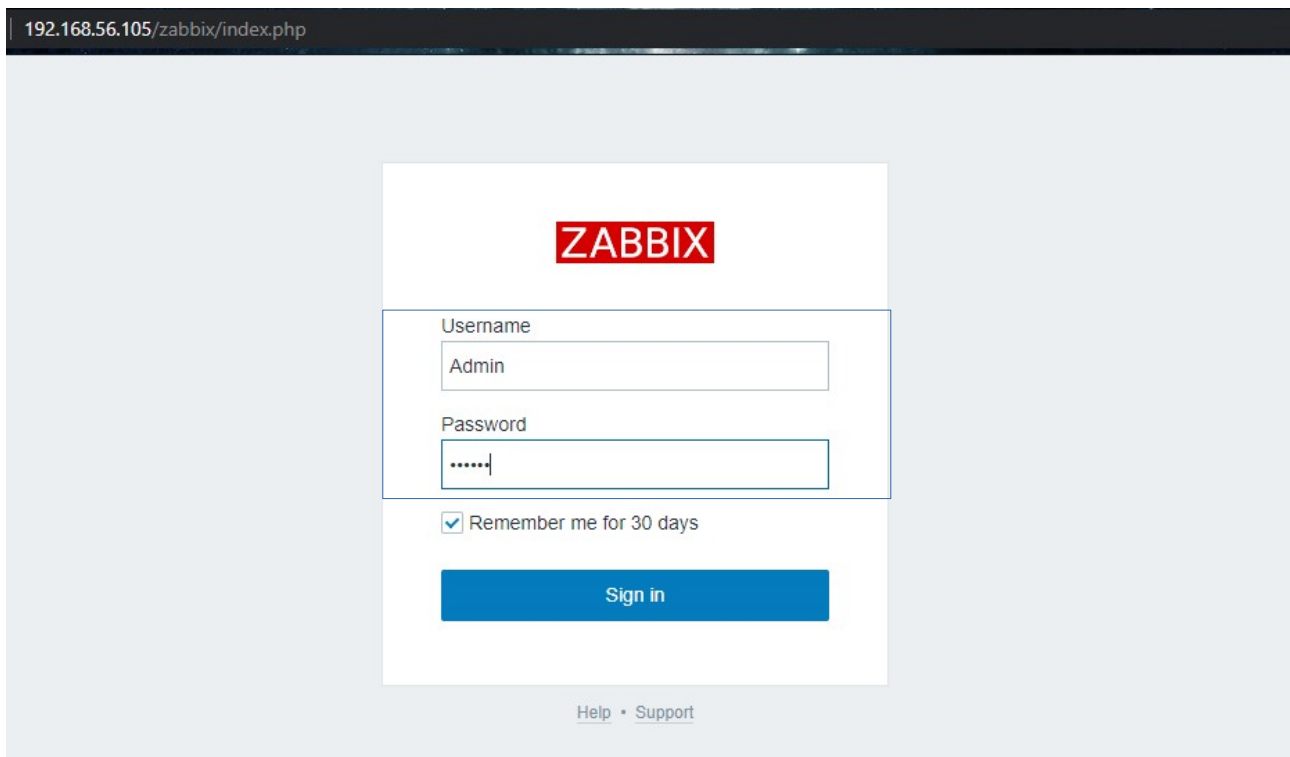
Database type	MySQL
Database server	localhost
Database port	default
Database name	zabbix
Database user	zabbix
Database password	*****
Database TLS encryption	false
Zabbix server	localhost
Zabbix server port	10051
Zabbix server name	Zabbix server

BackNext step

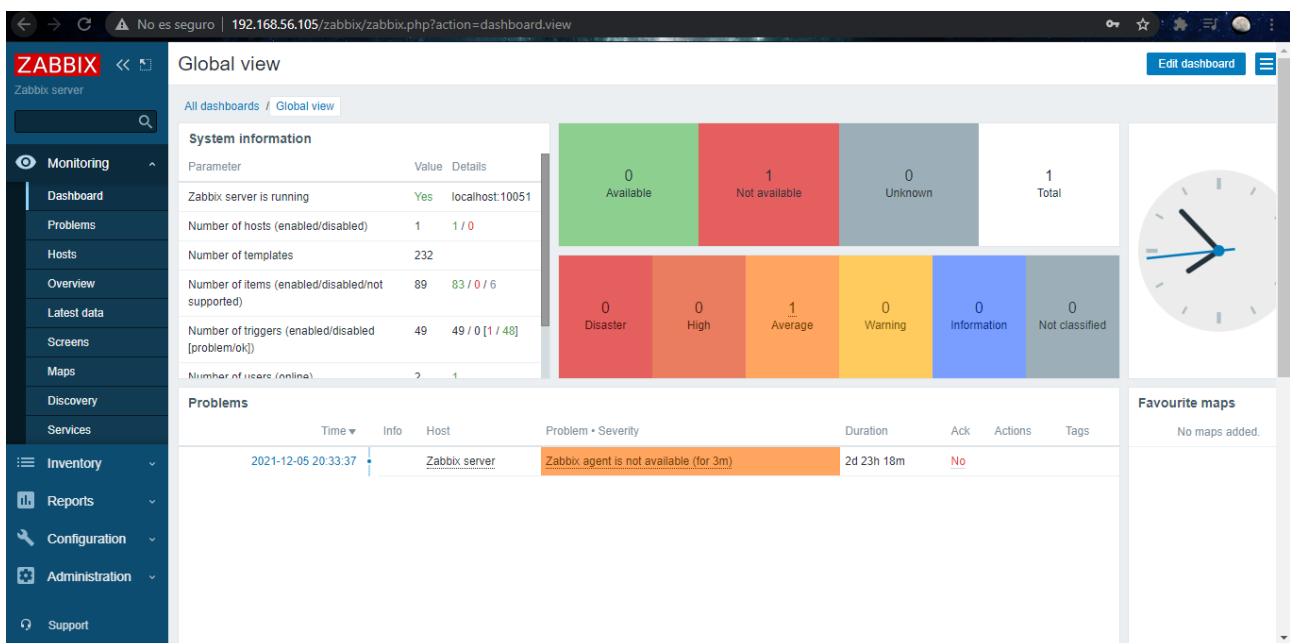
Licensed under [GPL v2](#)

*Datos correctos*

Una vez configurados y comprobados los datos necesarios para el inicio, accedemos como administrador introduciendo el usuario **'Admin'** y la contraseña **'zabbix'**.



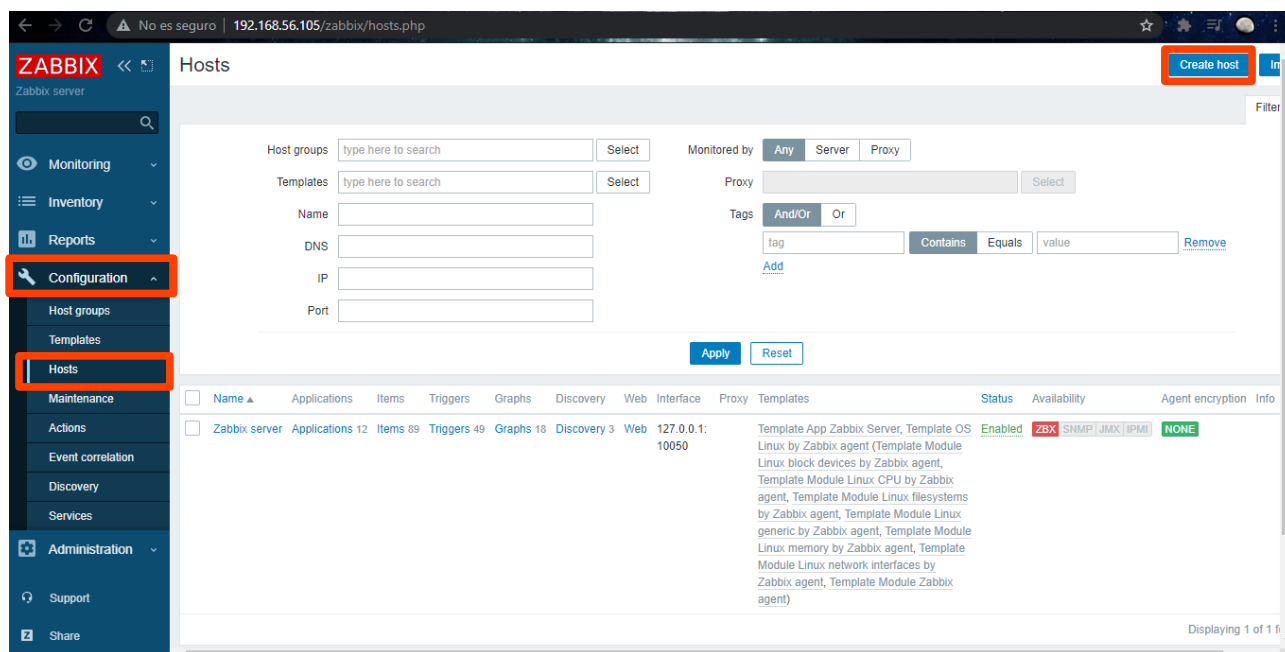
En la siguiente imagen se muestra el menú inicial que se muestra al entrar, donde se puede ver un panel de control con widgets relacionados con el sistema donde se aloja el servidor.



## 7.2. Configuración de la monitorización.

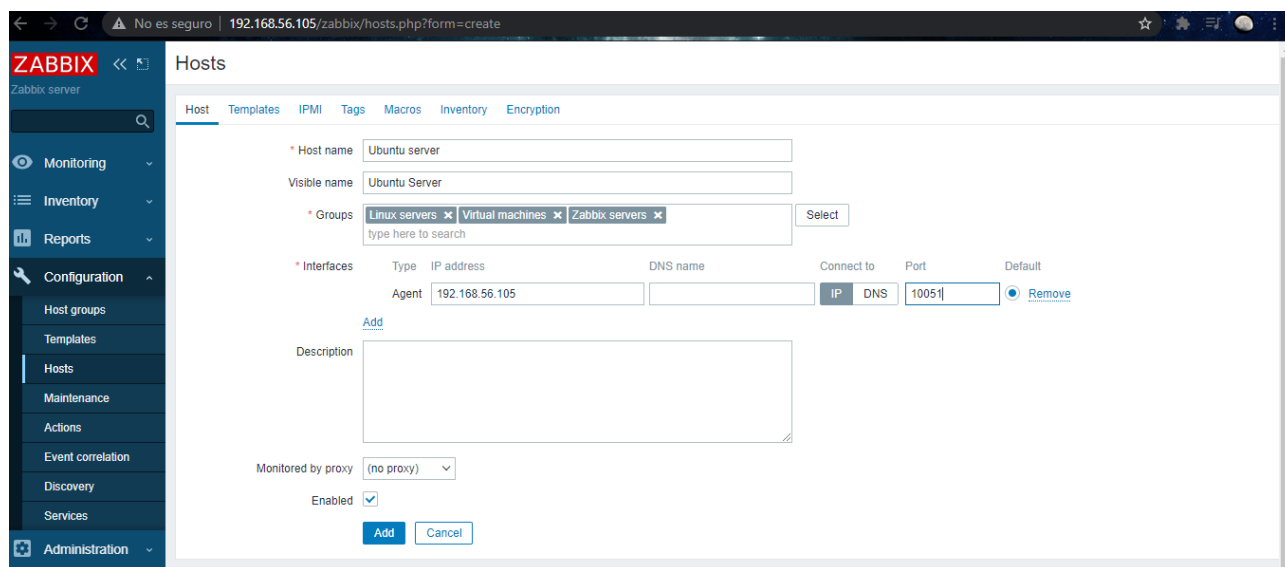
Antes de comenzar a monitorizar los servicios SSH y HTTP alojados en las máquinas, es necesario crear un host para cada una, donde habilitaremos dichos servicios.

### 7.2.1. Creación del host de Ubuntu Server.

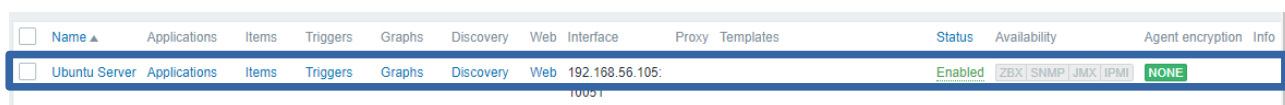


*Creación de un nuevo host*

Para Ubuntu Server, rellenamos los siguientes campos con los datos pertinentes.

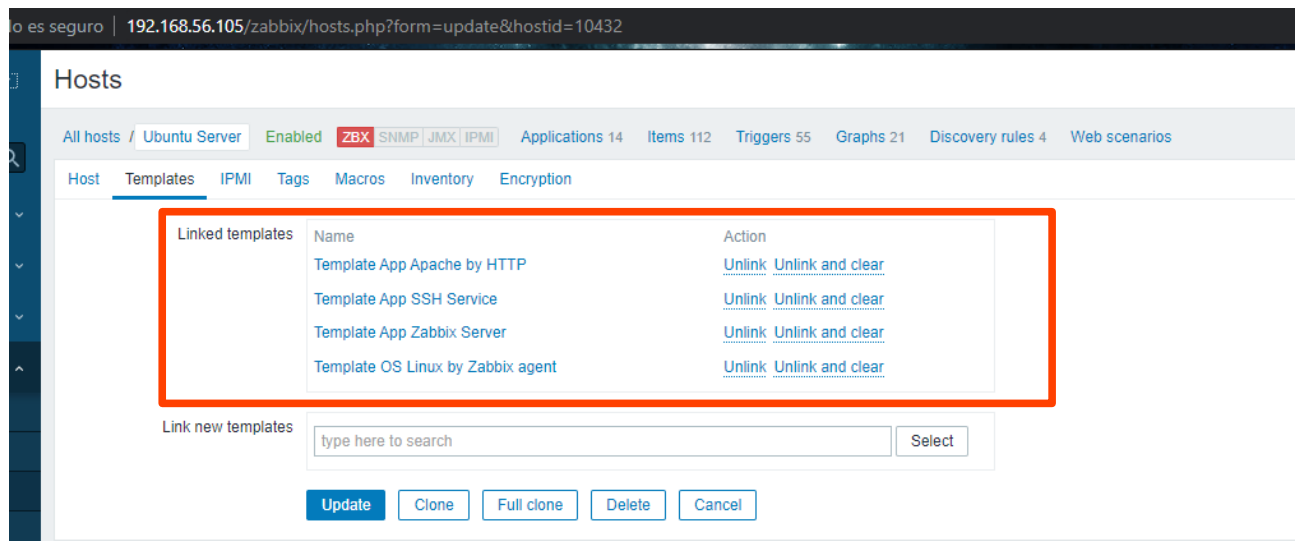


*Datos del host de Ubuntu Server*



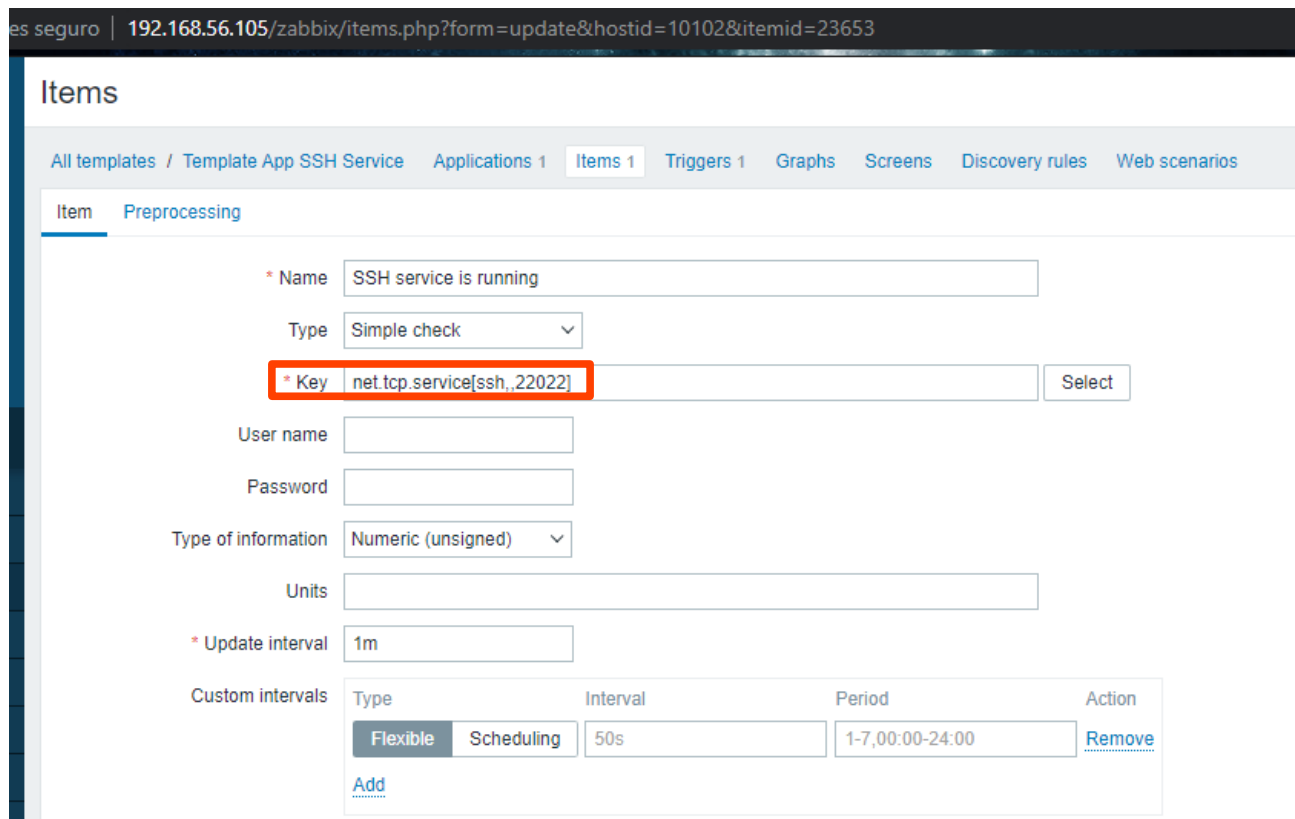
*Host creado*

Para poder monitorizar los servicios alojados en Ubuntu Server se debe modificar el host en ciertos aspectos. En primer lugar, debemos añadir una serie de *templates* al mismo. Estos *templates* son entidades que se pueden aplicar a múltiples hosts y se encuentran ordenados por grupos tal y como se muestra en el recuadro amarillo de la siguiente imagen.



*En azul, templates añadidos al host de Ubuntu Server*

El siguiente y último paso en la configuración del host es la modificación de la llave que indica el puerto del servicio SSH. Por defecto es el 22, por lo que tenemos que indicar que utilice el 22022. Para ello, nos situamos en el apartado de 'items' del host y procedemos con el cambio.





### 7.2.2. Creación del host de CentOS.

Para la creación del host de CentOS aplicaremos el mismo proceso que para el host de Ubuntu Server.

192.168.56.105/zabbix/hosts.php?form=create

### Hosts

Host Templates IPMI Tags Macros Inventory Encryption

\* Host name

Visible name

\* Groups    
type here to search

\* Interfaces

Type	IP address	DNS name	Connect to	Port	Default
Agent	<input type="text" value="192.168.56.110"/>	<input type="text"/>	<input checked="" type="button" value="IP"/> <input type="button" value="DNS"/>	<input type="text" value="10050"/>	<input checked="" type="radio"/> <input type="button" value="Remove"/>

[Add](#)

Description

Monitored by proxy

Enabled ☒

Datos del host de CentOS

<input type="checkbox"/>	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info
<input type="checkbox"/>	CentOs	Applications	Items	Triggers	Graphs	Discovery	Web	192.168.56.110:			Enabled	ZBX   SNMP   JMX   IPMI	NONE	

Host creado

192.168.56.105/zabbix/hosts.php?form=update&hostid=10434

### Hosts

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

Host Templates IPMI Tags Macros Inventory Encryption

Linked templates

Name	Action
Template App Apache by HTTP	<a href="#">Unlink</a> <a href="#">Unlink and clear</a>
Template App SSH Service	<a href="#">Unlink</a> <a href="#">Unlink and clear</a>
Template OS Linux by Zabbix agent	<a href="#">Unlink</a> <a href="#">Unlink and clear</a>

Link new templates

Templates

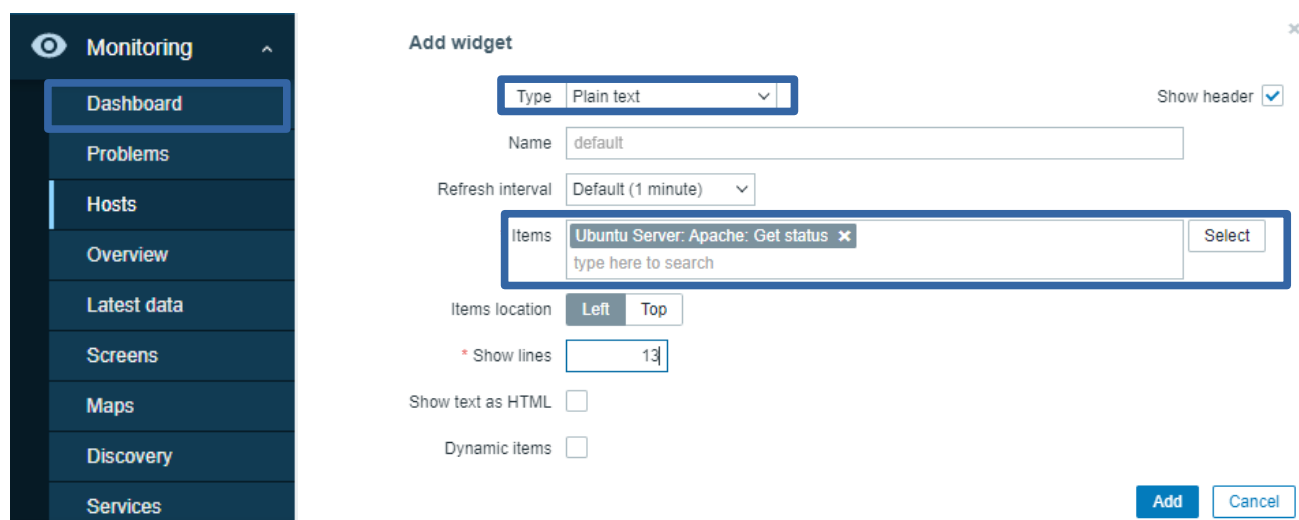
Name	Interface	Availability	Tags	Problems	Status	Latest data	Problems	Graphs	Screens	Web
CentOs	192.168.56.110: 10050	ZBX SNMP JMX IPMI		1	Enabled	Latest data	Problems 1	Graphs 16	Screens 3	Web
Ubuntu Server	192.168.56.105: 10051	ZBX SNMP JMX IPMI		1	Enabled	Latest data	Problems 1	Graphs 21	Screens 5	Web
Zabbix server	127.0.0.1: 10050	ZBX SNMP JMX IPMI		1	Enabled	Latest data	Problems 1	Graphs 18	Screens 4	Web

Hosts creados

## 8. Monitorización en Zabbix

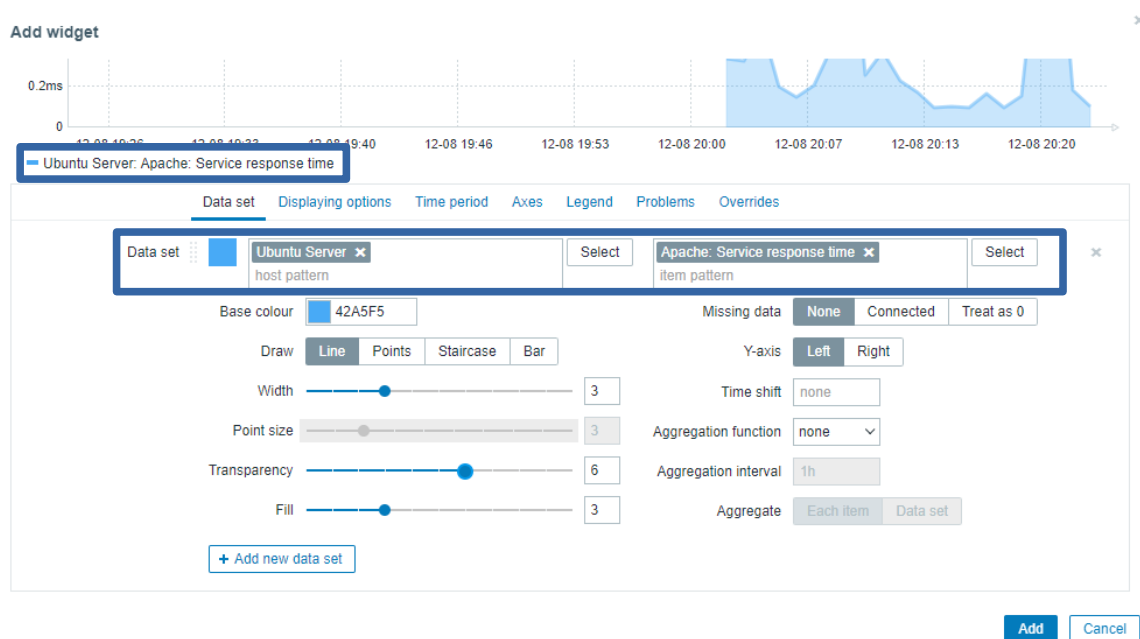
### 8.1. Ubuntu Server.

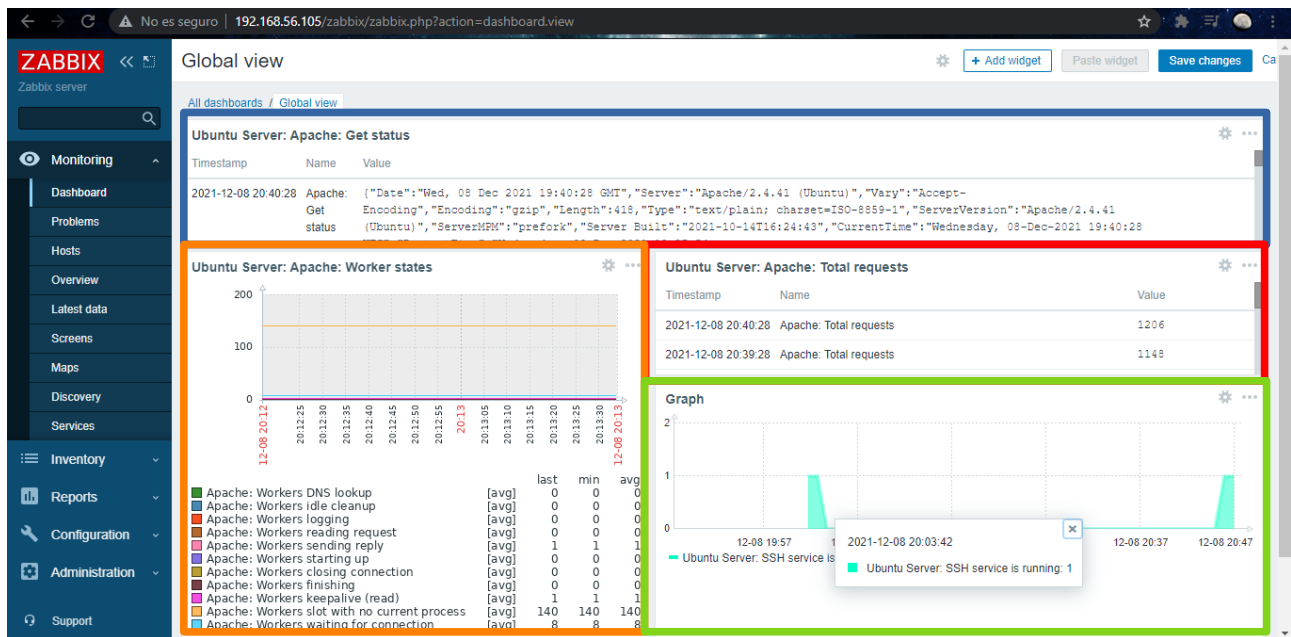
Para monitorizar el los servicios SSH y HTTP en Zabbix, nos situamos en el apartado ‘dashboards’, donde podremos colocar una serie de *widgets* que nos mostrarán la información que indiquemos, tal y como se muestra en las siguientes imágenes.



Valores de un widget

A cada *widget* debemos indicarle la información que mostrará y el modo de mostrarla. En el ejemplo de la imagen anterior podemos observar que mostrará el estado del servicio Apache en forma de texto plano. En la siguiente imagen vemos un ejemplo de creación de un *widget* que mostrará el tiempo de respuesta de Apache a modo de gráfico.





*Dashboard con widgets para monitorizar Ubuntu Server*

En la imagen anterior podemos observar un ejemplo de *dashboard* con varios *widget* que muestran información de los servicios de la máquina con Ubuntu Server. Concretamente, el estado de Apache (azul), el número total de peticiones a Apache (rojo), el estado de los trabajos de Apache (naranja) y, por último, un gráfico que indica si el servicio SSH se encuentra o no activo (verde).

## 8.2. CentOS.

Para CentOS, el proceso de creación de *widgets* es exactamente el mismo, por lo que no se mostrará dicho proceso. Sin embargo, para poder comenzar a monitorizar, debemos crear un item para SSH y otro para HTTP en el host de CentOS, tal y como se muestra en las imágenes siguientes. Esto es debido a que los servicios no se encuentran en la misma máquina que el servidor Zabbix, tal y como sucede en Ubuntu Server. En las imágenes de creación de items se mostrarán en azul los campos rellenados.

### Items

All hosts / CentOS Enabled ZBX SNMP JMX IPMI Applications 18 Items 98 Triggers 35 Graphs 18 Discovery rules 4 Web scenarios

Item Preprocessing

\* Name SSH service

Type Zabbix agent

\* Key net.tcp.service[ssh,192.168.56.110,22022] Select

\* Host interface 192.168.56.110 : 10050

Type of information Numeric (unsigned)

Units

\* Update interval 1m

Custom intervals

Type	Interval	Period	Action
Flexible	Scheduling	50s	1-7,00:00-24:00 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

New application

Item para SSH

Comprobamos que el item funciona. Debemos obtener el valor entero '1' para que esto sea así en el apartado 'test', antes de confirmar la creación del item.

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

Repetimos el mismo proceso para el ítem de HTTP, indicando en el key correspondiente, indicado en azul, el valor HTTP.

## Items

All hosts / CentOS Enabled **ZBX** SNMP JMX IPMI Applications 18 Items 99 Triggers 35 Graphs 18 Discovery rules 4 Web scenarios

Item Preprocessing

\* Name HTTP service

Type Zabbix agent

\* Key **net.tcp.service[http,192.168.56.110,22022]** Select

\* Host interface 192.168.56.110 : 10050

Type of information Numeric (unsigned)

Units

\* Update interval 1m

Custom intervals

Type	Interval	Period	Action
<b>Flexible</b> Scheduling	50s	1-7,00:00-24:00	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

New application

*Item para HTTP*

### Test item

Get value from host ☒

Host address 192.168.56.110 Port 10050

Proxy (no proxy)

Get value

Value 1 Time now

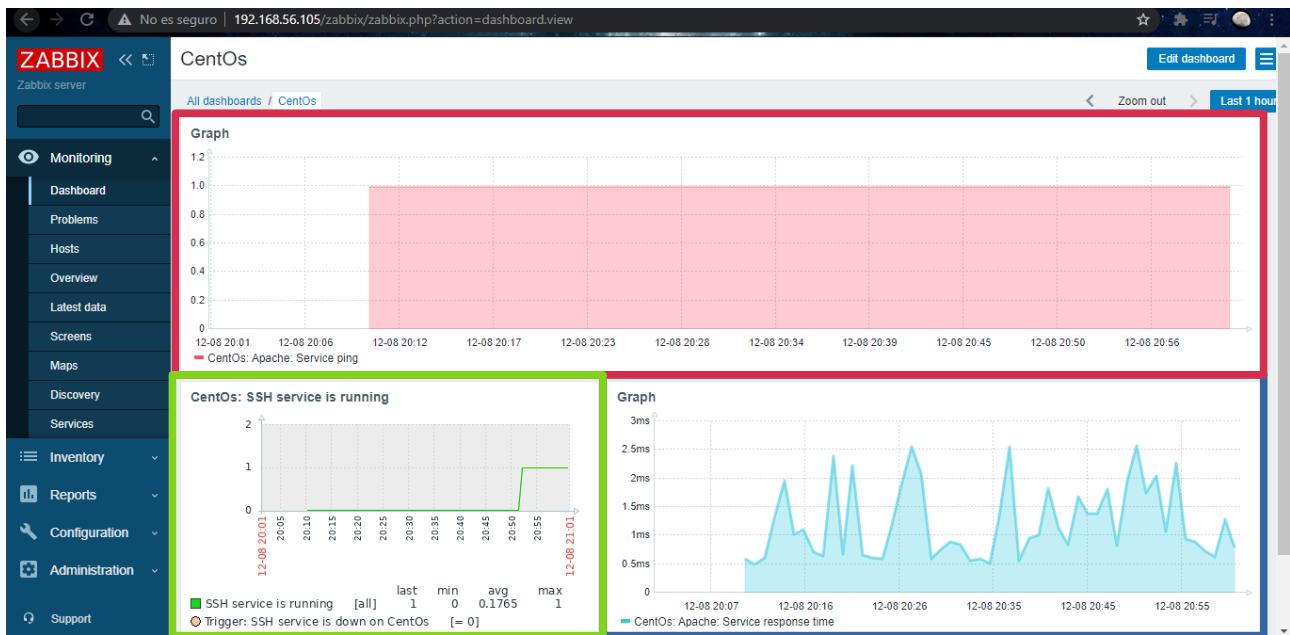
Previous value Prev. time

End of line sequence **LF** CRLF

Result Result converted to Numeric (unsigned) 1

Get value and test Cancel

*Prueba correcta*



En la imagen se muestra una *dashboard* con *widgets* que muestran información de los servicios de CentOS. Concretamente, se realiza un ‘ping’ al servicio Apache (rosa), se muestra el tiempo de respuesta de Apache (azul) y, tal y como se mostró en Ubuntu Server, un gráfico que indica el estado del servicio SSH (verde).

## 9. Ansible

Ansible es un servicio de automatización a nivel de plataforma. Puede utilizarse para ejecutar scripts o comandos, por ejemplo, desde una máquina en múltiples máquinas diferentes incluyendo al propio anfitrión donde se aloje Ansible.

En esta practica se muestra cómo instalar y configurar Ansible para poder utilizarlo así como algún que otro ejemplo de uso del mismo.

### 9.1. Instalación y configuración de Ansible en Ubuntu Server.

Para instalar Ansible en Ubuntu Server basta con ejecutar el comando siguiente:

```
alcaa@alcaa:~$ sudo apt install ansible
```

A continuación, debemos indicar qué hosts van a ser tratados con Ansible. Para ello, añadiremos las direcciones IP de los mismos en el fichero `/etc/ansible/hosts` tal y como se muestra en la imagen siguiente.

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
#
# Ex 1: Ungrouped hosts, specify before any group headers.
#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10
#
# 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
192.168.56.105
192.168.56.110
#
# 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]
-- INSERT --
```

El siguiente paso es modificar el puerto que utilizará Ansible. Por defecto utiliza el 22, por lo que debemos indicarle el 22022 en nuestro caso. Modificamos dicho valor en el fichero `/etc/ansible/ansible.cfg`

```
# config file for ansible -- https://ansible.com/
# =====

# nearly all parameters can be overridden in ansible-playbook
# or with command line flags. ansible will read ANSIBLE_CONFIG,
# ansible.cfg in the current working directory, .ansible.cfg in
# the home directory or /etc/ansible/ansible.cfg, whichever it
# finds first

[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

# plays will gather facts by default, which contain information about
# the remote system.
#
# smart - gather by default, but don't regather if already gathered
# implicit - gather by default, turn off with gather_facts: False
# explicit - do not gather by default, must say gather_facts: True
#gathering = implicit
-- INSERT --
```

1,1

Para poder hacer uso de las funciones de Ansible es necesario establecer una clave pública-privada que compartirán Ubuntu Server y CentOS.

```
alcaa@alcaa:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alcaa/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alcaa/.ssh/id_rsa
Your public key has been saved in /home/alcaa/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ho4XHAuYdaXnN60kCM/UgUY8hEg174wloi1fm5Co57g alcaa@alcaa
The key's randomart image is:
+----[RSA 3072]-----+
| ..0+=+00          |
| .+.+=0 .          |
| + +.*.0           |
| + 0 & B .         |
| + + 0 @ S + .     |
|..0 0 = 0 + 0      |
|..0 + 0 .          |
| + .               |
|E..                |
+----[SHA256]-----+
```



```
alcaa@alcaa:~$ ssh-copy-id 192.168.56.105 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/alcaa/.ssh/id_rsa"
The authenticity of host '[192.168.56.105]:22022 ([192.168.56.105]:22022)' can't be e
ECDSA key fingerprint is SHA256:ikaX1rFdLJA1G6oUxEW3H3FjqG8iXnI/fgRX1d5tteQ.
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 a
easily installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted no
all the new keys
alcaa@192.168.56.105's password:
Permission denied, please try again.
alcaa@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p '22022' '192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.
```

*Copia de la clave en Ubuntu Server*

```
alcaa@alcaa:~$ ssh-copy-id 192.168.56.110 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/alcaa/.ssh/id_rsa"
The authenticity of host '[192.168.56.110]:22022 ([192.168.56.110]:22022)' can't be e
ECDSA key fingerprint is SHA256:50rsIuCbWgsdYDF66nwp6YB7teq/b2Rz2GLbdgAbPv8.
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 a
easily installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted no
all the new keys
alcaa@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.
```

*Copia de la clave en CentOS*

Comprobamos a continuación que Ansible está funcionando correctamente de la siguiente forma:

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

*Ejecución del comando 'ping' desde Ubuntu Server a ambas máquinas con Ansible*

## 9.2. Ejemplos de uso de Ansible.

En las siguientes imágenes se muestran algunos ejemplos de uso de Ansible, así como el funcionamiento de los ‘playbooks’ que ofrece el servicio.

```
alcaa@alcaa:~$ ansible all -m command -a lsblk
192.168.56.105 | CHANGED | rc=0 >>
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                              7:0    0  55.5M  1 loop /snap/core18/2253
loop1                              7:1    0  67.2M  1 loop /snap/lxd/21835
loop2                              7:2    0   55M   1 loop /snap/core18/1880
loop3                              7:3    0  29.9M  1 loop /snap/snapd/8542
loop4                              7:4    0  42.2M  1 loop /snap/snapd/14066
loop5                              7:5    0  71.3M  1 loop /snap/lxd/16099
loop6                              7:6    0  61.9M  1 loop /snap/core20/1242
loop7                              7:7    0  61.9M  1 loop /snap/core20/1270
sda                                8:0    0   10G   0 disk
├─sda1                            8:1    0    1M   0 part
├─sda2                            8:2    0    1G   0 part /boot
├─sda3                            8:3    0    9G   0 part
└─ubuntu--vg-ubuntu--lv 253:0    0    9G   0 lvm  /
sr0                               11:0    1 1024M   0 rom
192.168.56.110 | CHANGED | rc=0 >>
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                8:0    0    8G   0 disk
├─sda1                            8:1    0    1G   0 part /boot
├─sda2                            8:2    0    7G   0 part
└─cl-root 253:0    0   6.2G   0 lvm  /
    └─cl-swap 253:1    0   820M   0 lvm  [SWAP]
sr0                               11:0    1 1024M   0 rom
```

Ejemplo1: Ejecución del comando ‘lsblk’ desde Ubuntu Server a ambas máquinas

```
alcaa@alcaa:~$ ansible -b -K -m user -a 'name=NuevoUsr' 192.168.56.110
BECOME password:
192.168.56.110 | CHANGED => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": true,
  "comment": "",
  "create_home": true,
  "group": 1001,
  "home": "/home/NuevoUsr",
  "name": "NuevoUsr",
  "shell": "/bin/bash",
  "state": "present",
  "system": false,
  "uid": 1001
}
```

```
alcaa@localhost ~$ ip --brief -color a
lo          UNKNOWN      127.0.0.1/8 ::1/128
enp0s3      UP              10.0.2.15/24 fe80::f3b6:3a32:ec35:4934/64
enp0s8      UP              192.168.56.110/24 fe80::a00:27ff:fe22:8646/64
alcaa@localhost ~$ ls /home
alcaa  NuevoUsr
```

Ejemplo2: Creación de un nuevo usuario desde Ubuntu Server en ambas máquinas

```
alcaa@alcaa:~$ sudo cat ejemplo.py
print("Ejecución de Script con Ansible realizada con éxito (Ubuntu Server)")
alcaa@alcaa:~$ ansible all -m command -a "python3 ejemplo.py"
192.168.56.105 | CHANGED | rc=0 >>
Ejecución de Script con Ansible realizada con éxito (Ubuntu Server)
192.168.56.110 | CHANGED | rc=0 >>
Ejecución de Script con Ansible realizada con éxito (Cnt0s)
```

```
alcaa@localhost ~$ sudo cat ejemplo.py
print("Ejecución de Script con Ansible realizada con éxito (Cnt0s)")
alcaa@localhost ~$
```

Ejemplo3: Ejecución de un script desde Ubuntu Server en ambas máquinas

## Playbooks.

Puede ocurrir que el número de acciones que queremos realizar en una máquina remota sea demasiado grande como para llevarlas a cabo de esta forma. Para solucionar este problema, existen los 'playbooks'. A continuación, se muestra la creación y ejecución de 'playbooks' en Ubuntu Server para CentOS.

Primero, se crean los directorios, si no están creados, que se observan en la imagen.

```
alcaa@alcaa:~$ sudo mkdir /etc/ansible/roles
[sudo] password for alcaa:
alcaa@alcaa:~$ sudo mkdir /etc/ansible/roles/basic
alcaa@alcaa:~$ sudo mkdir /etc/ansible/roles/basic/tasks
```

*Directorios para 'palybooks'*

Para poder ejecutar tareas en la máquina con CentOS, creamos en el directorio **/etc/ansible/roles/basic/tasks** el fichero **main.yml**.

```
alcaa@alcaa:~$ sudo vi /etc/ansible/roles/basic/tasks/main.yml
```

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

En este caso, el fichero ejecutara la tarea *Stop httpd*, que detendrá el servicio que le indiquemos tal y como se aprecia, en este ejemplo será httpd.

Creamos ahora el fichero **/etc/ansible/playbook.yml**.

```
alcaa@alcaa:~$ sudo vi /etc/ansible/playbook.yml
```

```
- hosts: 192.168.56.110
  become: true
  roles:
    - basic
~
```

Este dichero indica el host en el que se ejecutarán las tareas, en este caso CentOS. Además, la opción *become* definida a *true* nos permitirá ejecutar comandos como root. En *roles* se indican los roles que haya, para este ejemplo solo existe *basic*. Para ejecutar el 'playbook' procedemos como el la imagen.

```
alcaa@alcaa:~$ ansible-playbook -K /etc/ansible/playbook.yml
BECOME password:

PLAY [192.168.56.110] *****
TASK [Gathering Facts] *****
ok: [192.168.56.110]

TASK [basic : Stop http] *****
changed: [192.168.56.110]

PLAY RECAP *****
192.168.56.110 : ok=2  changed=1  unreachable=0  failed=0  skippe
0  ignored=0
```

```

Active: active (running) since Sat 2021-12-11 07:38:58 EST; 6s ago
Docs: man:httpd.service(8)
Main PID: 3731 (httpd)
Status: "Started, listening on: port 80"
Tasks: 213 (limit: 5019)
Memory: 24.3M
CGroup: /system.slice/httpd.service
├─3731 /usr/sbin/httpd -DFOREGROUND
├─3732 /usr/sbin/httpd -DFOREGROUND
├─3733 /usr/sbin/httpd -DFOREGROUND
├─3734 /usr/sbin/httpd -DFOREGROUND
└─3735 /usr/sbin/httpd -DFOREGROUND

dic 11 07:38:58 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 11 07:38:58 localhost.localdomain httpd[3731]: AH00558: httpd: Could not reliably
dic 11 07:38:58 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
dic 11 07:38:58 localhost.localdomain httpd[3731]: Server configured, listening on: p
falcaa@localhost ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: dis
  Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: inactive (dead) since Sat 2021-12-11 07:39:14 EST; 2s ago
     Docs: man:httpd.service(8)
  Process: 3731 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=
 Main PID: 3731 (code=exited, status=0/SUCCESS)
    Status: "Running, listening on: port 80"

dic 11 07:38:58 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
dic 11 07:38:58 localhost.localdomain httpd[3731]: AH00558: httpd: Could not reliably
dic 11 07:38:58 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
dic 11 07:38:58 localhost.localdomain httpd[3731]: Server configured, listening on: p
dic 11 07:39:13 localhost.localdomain systemd[1]: Stopping The Apache HTTP Server...
dic 11 07:39:14 localhost.localdomain systemd[1]: Stopped The Apache HTTP Server.

```

*Resultado de la ejecución del playbook anterior*

Con esta funcionalidad podemos llevar a cabo incontables acciones. Para no extender demasiado el contenido de esta memoria mostraré varios ejemplos simples de ‘playbooks’.

- Instalar GIT desde Ubuntu Server en CentOS:

```

- name: Stop http
  service: >
    name=httpd
    state=stopped
- name: Installing GIT
  yum:
    name=git
    state=latest

```

```

falcaa@falcaa:~$ ansible-playbook -K /etc/ansible/playbook.yml
BECOME password:

PLAY [192.168.56.110] *************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.110]

TASK [basic : Stop http] *************************************************************
ok: [192.168.56.110]

TASK [basic : Installing GIT] *************************************************************
changed: [192.168.56.110]

PLAY RECAP *********************************************************************
192.168.56.110      : ok=3    changed=1    unreachable=0    failed=0    skippe
0 ignored=0

```

- Transferir un archivo desde Ubuntu Server a CentOS:

```
alcaa@alcaa:~$ sudo touch archivo.cc
alcaa@alcaa:~$ ls
archivo.cc  ejemplo.py  zabbix-release_5.0-1+focal_all.deb
```

*Archivo a transferir*

```
- name: Transfer file
  copy:
    src: /home/alcaa/archivo.cc
    dest: /home/alcaa
    owner: root
    mode: '0644'
```

```
alcaa@alcaa:~$ ansible-playbook -K /etc/ansible/playbook.yml
BECOME password:
PLAY [192.168.56.110] *****
TASK [Gathering Facts] *****
ok: [192.168.56.110]
TASK [basic : Transfer file] *****
changed: [192.168.56.110]
PLAY RECAP *****
192.168.56.110      : ok=2    changed=1    unreachable=0    failed=0    skippe
0 ignored=0

[alcaa@localhost ~]$ ls
ejemplo.py
[alcaa@localhost ~]$ ls
archivo.cc  ejemplo.py
[alcaa@localhost ~]$
```

*Resultado de la transferencia*

## 10. Referencias

### Zabbix

- Documentación de Zabbix 5.0: [https://www.zabbix.com/documentation/5.0/en/manual/installation/install\\_from\\_packages](https://www.zabbix.com/documentation/5.0/en/manual/installation/install_from_packages)
- Documentación de Zabbix 5.0: <https://www.zabbix.com/documentation/5.0/en/manual/concepts>
- [Vadym Kalsin] digitalocean.com: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-zabbix-to-securely-monitor-remote-servers-on-ubuntu-20-04-es>
- ZabbixMan: <https://soka.gitlab.io/ZabbixMan/server/problemas/>

### Ansible

- Documentación de Ansible: <https://docs.ansible.com/ansible/latest/cli/ansible-playbook.html>
- [Avi] geekflare.com: <https://geekflare.com/es/ansible-installation-ubuntu/>
- [Stephen Rees-Carter, Mark Drake, Erika Heidi] digitalocean.com: <https://www.digitalocean.com/community/tutorials/como-instalar-y-configurar-ansible-en-ubuntu-18-04-es>
- [Erika Heidi] digitalocean.com: <https://www.digitalocean.com/community/cheatsheets/how-to-use-ansible-cheat-sheet-guide>