

INGENIERÍA DE SERVIDORES (2021-2022)
GRADO EN INGENIERÍA INFORMÁTICA
UNIVERSIDAD DE GRANADA

Memoria Práctica 3

Alberto Llamas González

4 de enero de 2022

Índice

1 Ejercicio 1	3
1.1 Instalación y configuración de Zabbix en Ubuntu Server	3
1.1.1 Configuración de la base de datos	4
1.1.2 Configuración de php para la interfaz de Zabbix	6
1.1.3 Configuración del agente de Zabbix	7
1.2 Instalación y configuración de Zabbix Agent en CentOS	8
1.2.1 Configuración del agente de Zabbix	9
1.3 Configuración del frontend	11
1.4 Configuración de la monitorización de los servicios SSH y HTTP	13
2 Ejercicio 2	18
2.1 Instalación y configuración de Ansible en Ubuntu Server	18

1. Ejercicio 1

Realice una instalación de Zabbix 5.0 en su servidor con Ubuntu Server 20.04 y configure para que se monitorice a él mismo y para que monitorice a la máquina con CentOS. Puede configurar varios parámetros para monitorizar, uso de CPU, memoria, etc. pero debe configurar de manera obligatoria la monitorización de los servicios SSH y HTTP.

1.1. Instalación y configuración de Zabbix en Ubuntu Server

Para instalar Zabbix debemos descargarnos el repositorio Zabbix desde su página de descargas. Para evitar errores, me he conectado al servidor UbuntuServer por SSH desde el host, permitiéndome copiar los comandos de la página de Zabbix. Una vez descargado, descomprimiremos el paquete con la opción **dpkg -i** (-i para instalar el paquete). Para ello ejecutamos los siguientes comandos:

```
1 $ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/
2 zabbix-release/zabbix-release_5.0-1+focal_all.deb
3 $ dpkg -i zabbix-release_5.0-1+focal_all.deb
```

```
albertolg@ubuntu:~$ sudo systemctl restart sshd.service
albertolg@ubuntu:~$ sudo wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
--2021-12-26 14:07:04-- 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::20
62: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- 100%[=====] 4.14K --.-KB/s in 0s
2021-12-26 14:07:05 (340 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [424/4244]

albertolg@ubuntu:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
dpkg: warning: downgrading zabbix-release from 1:5.4-1+ubuntu20.04 to 1:5.0-1+focal
(Reading database ... 71164 files and directories currently installed.)
Preparing to unpack zabbix-release_5.0-1+focal_all.deb ...
Unpacking zabbix-release (1:5.0-1+focal) over (1:5.4-1+ubuntu20.04) ...
Setting up zabbix-release (1:5.0-1+focal) ...
Installing new version of config file /etc/apt/sources.list.d/zabbix.list ...
```

Para acabar de instalar el repositorio actualizamos la lista de paquetes

```
1 $ sudo apt update
```

```
albertolg@ubuntu:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Get:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4958 B]
Get:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1190 B]
Get:7 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [3226 B]
]
Fetched 9374 B in 1s (8523 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
214 packages can be upgraded. Run 'apt list --upgradable' to see them.
albertolg@ubuntu:~$
```

Para acabar, instalamos el servidor, la interfaz y el agente de Zabbix.

```
1 $ sudo apt install zabbix-server-mysql zabbix-frontend-php
2 zabbix-apache-conf zabbix-agent
```

1.1.1. Configuración de la base de datos

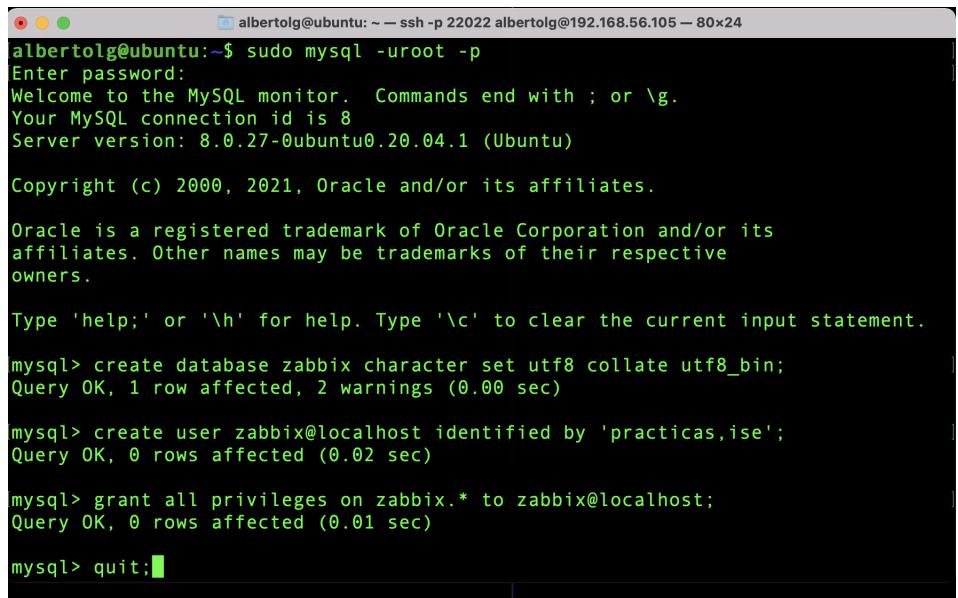
Aquí nos encontramos con el siguiente problema.

```
albertolg@ubuntu:~$ mysql -uroot -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run
/mysqld/mysqld.sock' (2)
albertolg@ubuntu:~$ sudo mysql -uroot -p
Enter password:
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run
/mysqld/mysqld.sock' (2)
albertolg@ubuntu:~$ sudo systemctl status mysql.service
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset:enabled)
  Active: inactive (dead)
albertolg@ubuntu:~$
```

Para solucionarlo, me dí cuenta de que on tenía el servicio de mysql activo por lo que ejecutando

```
1 $ sudo systemctl start mysql.service
```

lo solucioné. Continuamos con la configuración de la base de datos entrando como root donde la contraseña va a ser 'practicas,ise' (aunque lo más seguro es no repetir contraseñas), creamos la base de datos y un usuario **zabbix** que se identifica con la contraseña 'practicas,ise' al cual le damos la característica de super usuario.



```
albertolg@ubuntu:~$ sudo mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.27-0ubuntu0.20.04.1 (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 (0.00 sec)

mysql> create user zabbix@localhost identified by 'practicas,ise';
Query OK, 0 rows affected (0.02 sec)

mysql> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> quit;
```

Importamos ahora en el servidor Zabbix, el esquema y los datos iniciales. Esto tarda unos minutos y no deberíamos interrumpirlo ya que fallaría Zabbix.

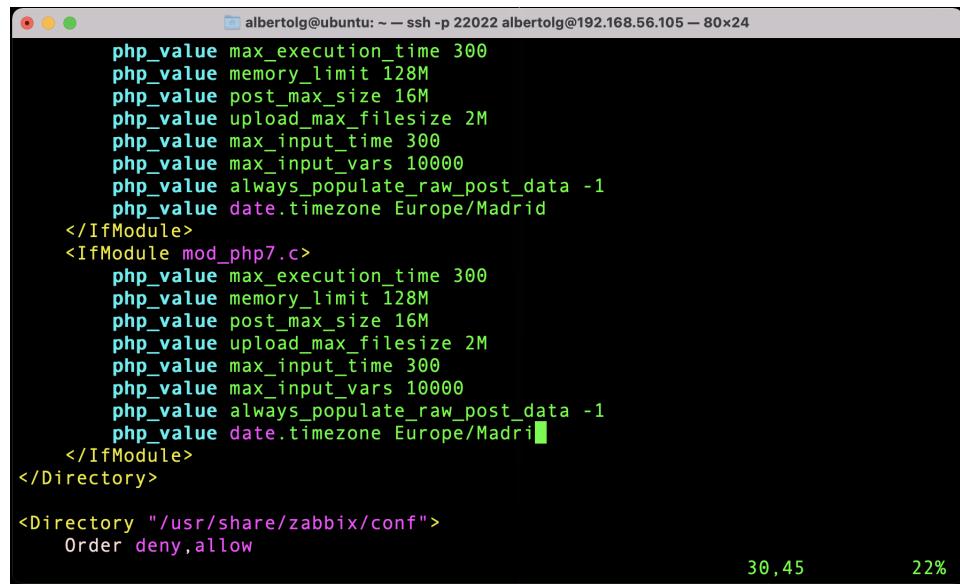
```
1 $ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz |  
2 mysql -uzabbix -p zabbix
```

Configuramos la base de datos de Zabbix modificando el archivo **/etc/zabbix/zabbix_server.conf**, donde únicamente tenemos que cambiar la DBPassword a la contraseña que hayamos puesto a la hora de la configuración con **mysql** en nuestro caso

```
$ DBPassword=practicas ,ise
```

1.1.2. Configuración de php para la interfaz de Zabbix

Configuramos PHP para el frontend en `/etc/zabbix/apache.conf` y descomentamos la zona horaria de la región y cambiamos Riga por nuestra zona horaria en España (Madrid).



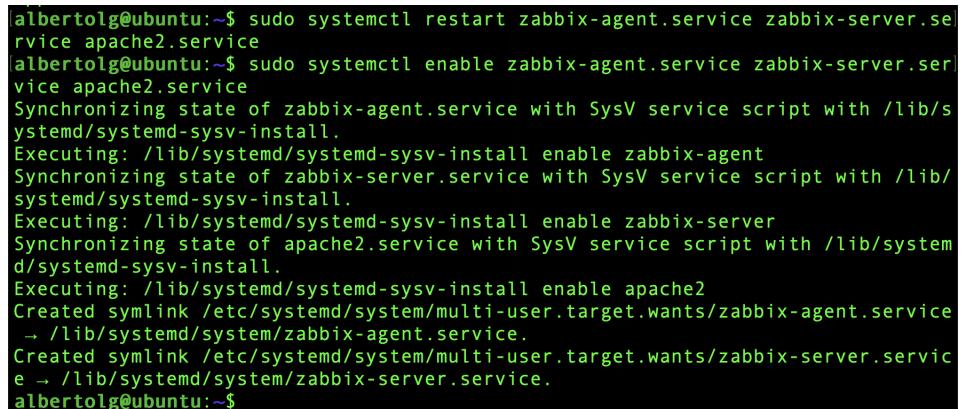
```
albertolg@ubuntu: ~ - ssh -p 22022 albertolg@192.168.56.105 - 80x24
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value max_input_vars 10000
php_value always_populate_raw_post_data -1
php_value 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/Madri
</IfModule>
</Directory>

<Directory "/usr/share/zabbix/conf">
    Order deny,allow

```

30,45 22%

Una vez lo hayamos configurado, reiniciamos y habilitamos **zabbix-server**, **zabbix-agent** y **apache2**



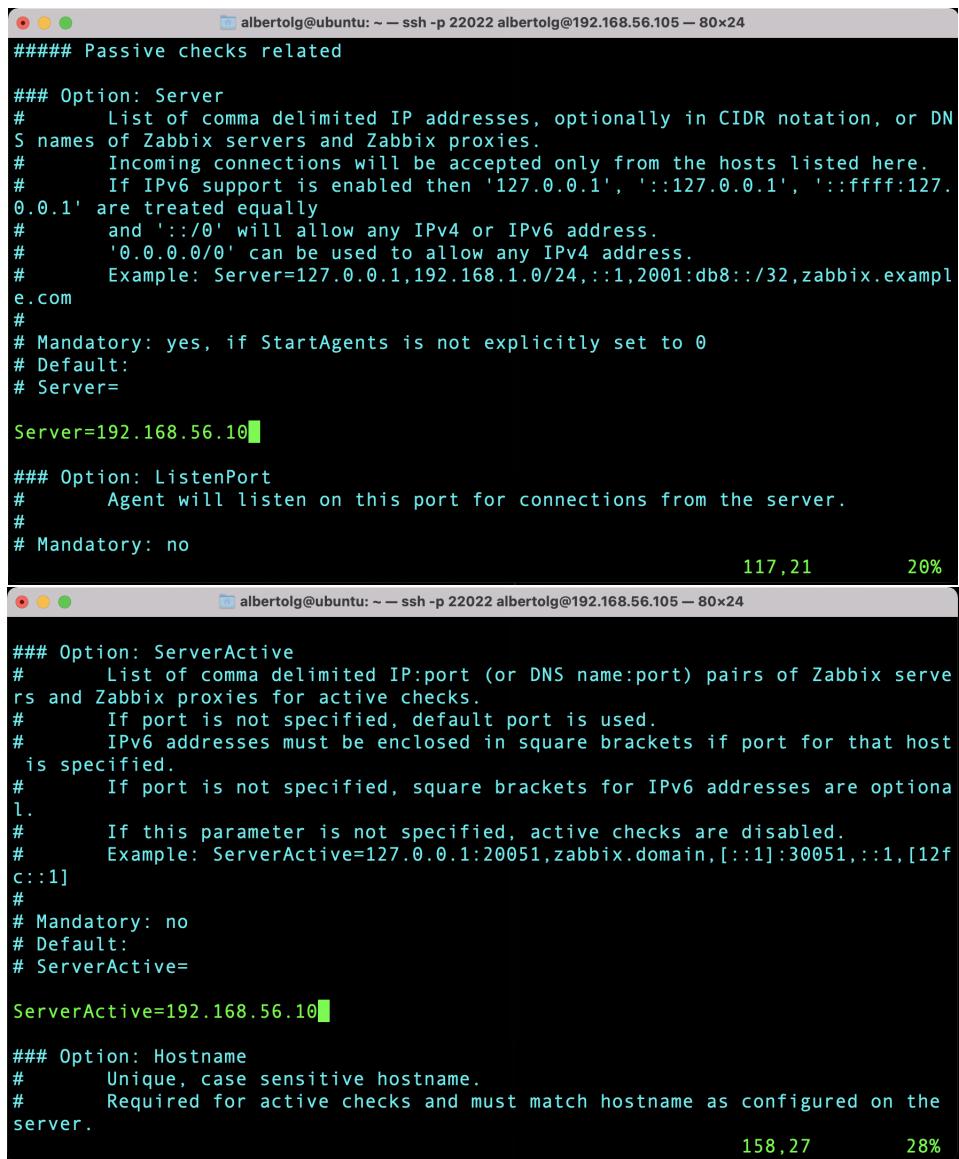
```
albertolg@ubuntu:~$ sudo systemctl restart zabbix-agent.service zabbix-server.servi
rvce apache2.service
albertolg@ubuntu:~$ sudo systemctl enable zabbix-agent.service zabbix-server.servi
rvce apache2.service
Synchronizing state of zabbix-agent.service with SysV service script with /lib/s
ystemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
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/system
d/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service
→ /lib/systemd/system/zabbix-agent.service.
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.servic
e → /lib/systemd/system/zabbix-server.service.
albertolg@ubuntu:~$
```

Habilitamos ahora los puertos de escucha de Zabbix ya que el cortafuegos de UbuntuServer bloquea por defecto todos los puertos con

```
1 $ sudo ufw allow 10050/tcp
2 $ sudo ufw allow 10051/tcp
```

1.1.3. Configuración del agente de Zabbix

Para configurar ahora el agente de Zabbix, accedemos al archivo `etc/zabbix/zabbix_agent.conf` donde indicamos la IP del servidor, 192.168.56.105 en nuestro caso, tanto en Server como en ServerActive permitiéndonos así conectarnos desde fuera.



```
albertolg@ubuntu: ~ - ssh -p 22022 albertolg@192.168.56.105 - 80x24
##### Passive checks related

### Option: Server
#      List of comma delimited IP addresses, optionally in CIDR notation, or DN
S 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.exampl
e.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.56.10

### Option: ListenPort
#      Agent will listen on this port for connections from the server.
#
# Mandatory: no
117,21          20%


albertolg@ubuntu: ~ - ssh -p 22022 albertolg@192.168.56.105 - 80x24

### Option: ServerActive
#      List of comma delimited IP:port (or DNS name:port) pairs of Zabbix serve
rs 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 optiona
l.
#      If this parameter is not specified, active checks are disabled.
#      Example: ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12f
c::1]
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.56.10

### Option: Hostname
#      Unique, case sensitive hostname.
#      Required for active checks and must match hostname as configured on the
server.
158,27          28%
```

Una vez hecho esto, solo queda reiniciar los servicios `zabbix-server`, `zabbix-agent` y `apache2` y comprobar que están activos.

1.2. Instalación y configuración de Zabbix Agent en CentOS

Para instalar Zabbix agent en CentOS, elegimos en la página de descargas de Zabbix el OS CentOS y obtenemos el repositorio a descargar.

```
1 $ rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-
     release-5.0-1.el8.noarch.rpm
2 $ dnf clean all
```

```
[albertolg@localhost ~]$ sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/
/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
[albertolg@localhost ~]$ [sudo] password for albertolg:
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.nNKt4i: EncabezadoV4 RSA/SHA512 Signature, ID de cl
ave a14fe591: NOKEY
Verificando...                               #####[100%]
Preparando...                                #####[100%]
Actualizando / instalando...
 1:zabbix-release-5.0-1.el8                #####[100%]
[albertolg@localhost ~]$ dnf clean all
0 archivos eliminados
[albertolg@localhost ~]$ █
```

Instalamos únicamente el agente de Zabbix

```
[albertolg@localhost ~]$ sudo dnf install zabbix-agent
Última comprobación de caducidad de metadatos hecha hace 0:00:06, el dom 26 dic
2021 09:35:06 EST.
Dependencias resueltas.
=====
Paquete           Arquitectura   Versión       Repositorio   Tam.
=====
Instalando:
zabbix-agent      x86_64         5.0.19-1.el8  zabbix        490 k

Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total de la descarga: 490 k
Tamaño instalado: 2.1 M
[;Está de acuerdo [s/N]? s
Descargando paquetes:
zabbix-agent-5.0.19-1.el8.x86_64.rpm      324 kB/s | 490 kB   00:01
-----
Total            324 kB/s | 490 kB   00:01
advertencia:/var/cache/dnf/zabbix-b7349cbb4866b08d/packages/zabbix-agent-5.0.19-
1.el8.x86_64.rpm: EncabezadoV4 RSA/SHA512 Signature, ID de clave a14fe591: NOKEY
Zabbix Official Repository - x86_64          1.6 MB/s | 1.7 kB   00:00

Instalado:
  zabbix-agent-5.0.19-1.el8.x86_64

¡Listo!
[albertolg@localhost ~]$
```

1.2.1. Configuración del agente de Zabbix

Para configurar el agente, al igual que en UbuntuServer debemos habilitar el puerto ya que el firewall de CentOS los cierra.

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

Una vez habilitado el puerto, editamos el archivo **/etc/zabbix/zabbix_agentd.conf** indicando la IP el servidor Zabbix en este caso, la IP de UbuntuServer.

```
UbuntuExamenP2 [Running]
albertolg@ubuntu:~$ sudo systemctl restart sshd.service
albertolg@ubuntu:~$ 

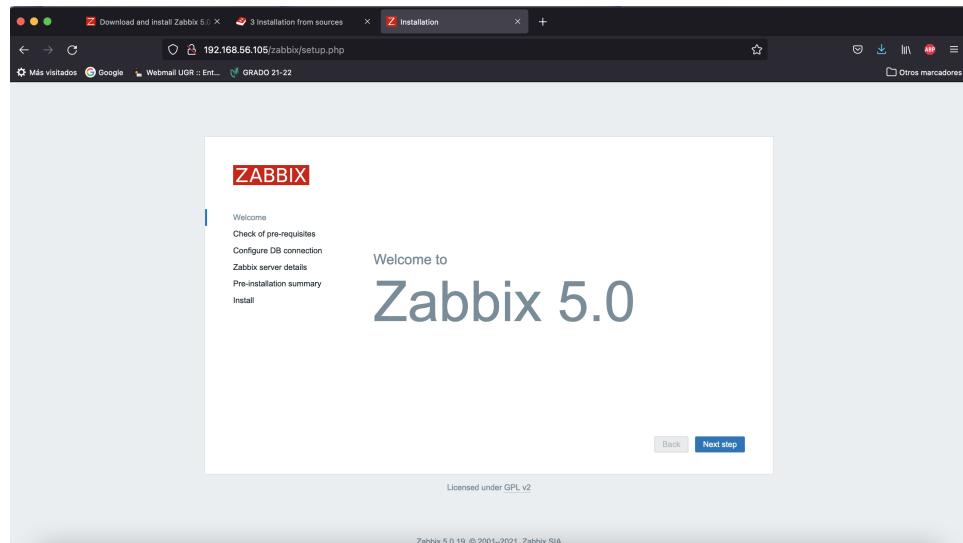
# 
# List of comma delimited IP addresses, optionally in CIDR notation, or DN
# names of Zabbix servers and Zabbix proxies.
# Incoming connections will be accepted only from the hosts listed here.
# If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.
# 0.0.1' and '::/0' will allow any IPv4 or IPv6 address.
# '0.0.0.0/0' can be used to allow any IPv4 address.
# Example: Servers=127.0.0.1,192.168.1.0/24,:1,2001:db8::/32,zabbix.exampl
e.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=
#
Server=192.168.56.105
### Option: ListenPort
# Agent will listen on this port for connections from the server.
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10050
-- INSERT --

CentOSExamenP2 [Running]
albertolg@localhost:~$ 
albertolg@localhost:~$ albertolg@localhost:~$ ssh -p 22022 albertolg@192.168.56.110 -80x24
# StartAgents=3
#####
# Active checks related
#####
# List of comma unlimited IP:port (or DNS name:port) pairs of Zabbix serve
rs and proxy addresses 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,[2f
c::1]
#
# Mandatory: no
# Default:
# ServerActive=
#
ServerActive=192.168.56.105
### Option: Hostname
-- INSERT --
```

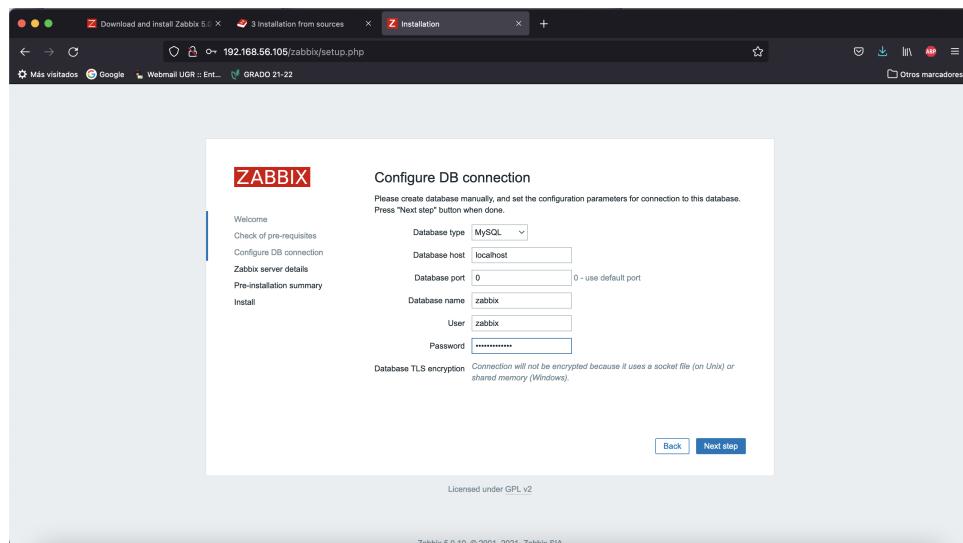
Por último, activamos el agente

1.3. Configuración del frontend

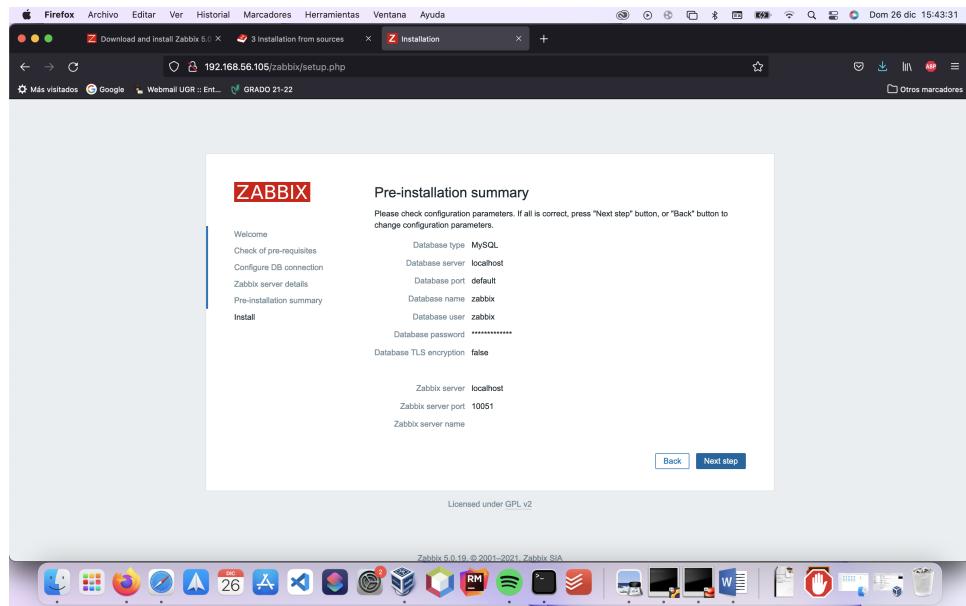
Podemos configurar el frontend de Zabbix accediendo desde un navegador a la dirección IP/zabbix, en nuestro caso **192.168.56.105/zabbix**. Una vez dentro, vamos avanzando las páginas e introducimos en la página de configuración de la conexión a la BD de MySQL, la contraseña que le habíamos puesto, 'practicas,ise' y finalizamos la instalación.



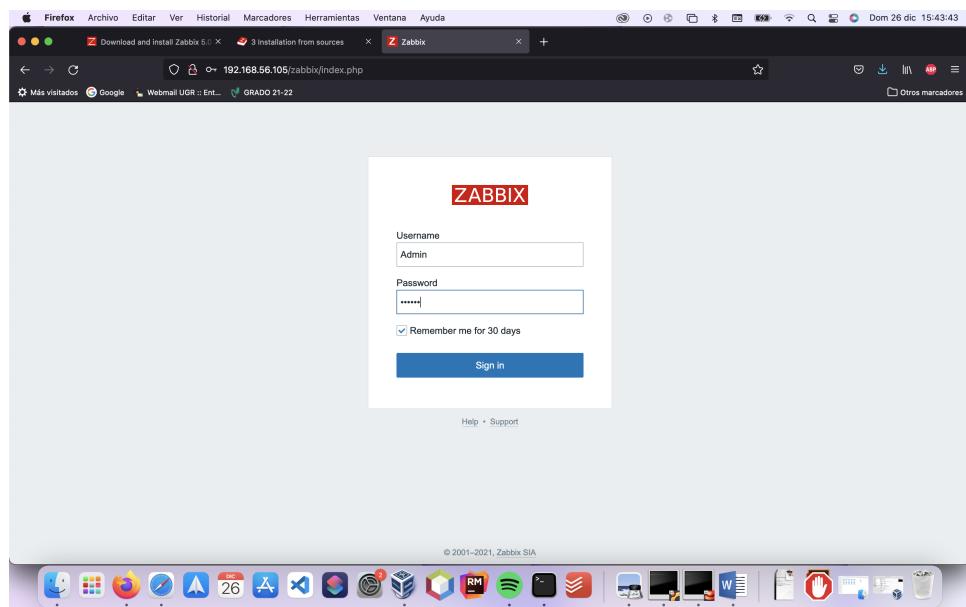
Nos conectamos a la BD introduciendo la contraseña que habíamos puesto antes en la configuración, 'practicas,ise'



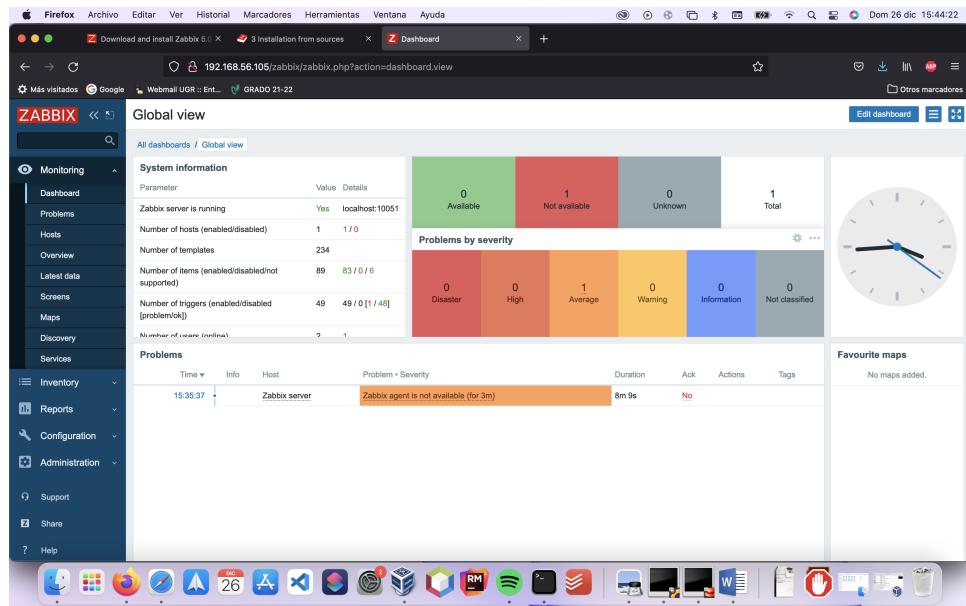
Comprobamos que lo introducido es correcto



Una vez acabado, iniciamos sesión como administrado en la página a la que nos redirige **192.168.56.105/zabbix/index.php** con el usuario **Admin** y contraseña **zabbix**



Cuando estemos dentro, podremos monitorizar el servidor de Ubuntu



1.4. Configuración de la monitorización de los servicios SSH y HTTP

Para monitorizar los servicios SSH y HTTP tanto de UbuntuServer como de CentOS, es necesario que creamos un host para cada uno de ellos. Nos vamos a Configuration >Hosts >Create host. Rellenamos los siguientes campos:

A screenshot of a Firefox browser window showing the 'Configuration > Hosts > Create host' form. The sidebar on the left lists Hosts, Templates, IPMI, Tags, Macros, Inventory, and Encryption. The main form has tabs for Host, Templates, IPMI, Tags, Macros, Inventory, and Encryption. Under the Host tab, fields include 'Host name' (UbuntuServer), 'Visible name' (UbuntuServer), 'Groups' (Linux servers, Virtual machines, Zabbix servers), 'Interfaces' (Agent IP address 192.168.56.105, Port 10550, Connect to IP), 'Description' (empty), 'Monitored by proxy' (no proxy), and 'Enabled' (checkbox checked). Buttons at the bottom are 'Add' and 'Cancel'. The status bar at the bottom indicates 'Zabbix 5.0.19. © 2001–2021, Zabbix SIA'.

En CentOS, sería igual salvo la IP address que sería **192.168.56.110**.

Activamos ahora ambos Hosts y nos queda lo siguiente

The screenshot shows the Zabbix web interface with two hosts listed:

- UbuntuServer**: Applications 11, Items 42, Triggers 14, Graphs 8, Discovery 3, Web 192.168.56.105:10050. Status: Enabled. Monitored by ZBX, SNMP, JMX, IPMI. Agent encryption: NONE. Description: Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)
- CentOS**: Applications 11, Items 42, Triggers 14, Graphs 8, Discovery 3, Web 192.168.56.110:10050. Status: Enabled. Monitored by ZBX, SNMP, JMX, IPMI. Agent encryption: NONE. Description: Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)

Debemos ahora añadir los templates de monitorización a Ubuntu Server y a CentOS. Muestro únicamente Ubuntu Server pero se hace idéntico para CentOS.

The screenshot shows the Firefox browser displaying the Zabbix configuration interface for host templates. The URL is 192.168.56.105/zabbix/hosts.php?form=update&hostid=10434. The page shows the following details for the host template:

Name	Action
Template App HTTP Service	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

Below the table, there is a search bar labeled "Link new templates" with the placeholder "type here to search". At the bottom, there are buttons for "Update", "Clone", "Full clone", "Delete", and "Cancel".

Si recordamos de la práctica 2, para dar seguridad al SSH modificamos el puerto al 22022 por lo que debemos modificar eso en Zabbix. Pulsamos en el host de Ubuntu Server >Template App SSH Service

Pulsamos en SSH service is running y nos vamos a Items. Una vez en la pestaña de Items, modificamos la key y escribimos **net.tcp.service[ssh,,22022]**. No sería necesario repetir el proceso en el Host de CentOS ya que hemos modificado el template por lo que se aplicará a todos los hosts que creemos que tengan dicho template.

Ya estaría todo configurado para poder monitorizar SSH y HTTP. Por último, nos vamos a Monitoring >Dashboard >Add widget. En mi caso, para poder ver claro que monitoriza correctamente tanto Ubuntu Server como CentOS, he añadido 4 widgets, 2 para monitorizar SSH de los respectivos servidores virtuales y 2 para monitorizar HTTP.

Para comprobar que los monitoriza correctamente he pausado el servicio SSH en Ubuntu Server. Como vemos en el widget, nos lo monitoriza correctamente.

```
alberto@ubuntu:~$ sudo systemctl stop ssh.service
alberto@ubuntu:~$ sudo systemctl status ssh.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun Dec 26 16:38:48 UTC; 2s ago
     Docs: man:sshd(8)
           man:systemctl(1)
             Process: 2897 ExecStart=/usr/sbin/sshd -D $SSH_OPTS (code=exited, status=0)
             Main PID: 2897 (code=exited, status=0/SUCCESS)
Dec 26 15:33:09 ubuntu sshd[18893]: Connection closed by 192.168.56.185 port 56
Dec 26 15:34:09 ubuntu sshd[18811]: Connection closed by 192.168.56.185 port 56
Dec 26 15:34:09 ubuntu sshd[18890]: Connection closed by 192.168.56.185 port 56
Dec 26 15:36:09 ubuntu sshd[18951]: Connection reset by 192.168.56.185 port 56
Dec 26 15:36:09 ubuntu sshd[18950]: Connection closed by 192.168.56.185 port 56
Dec 26 15:38:09 ubuntu sshd[18990]: Connection closed by 192.168.56.185 port 56
Dec 26 15:38:48 ubuntu systemd[1]: Stopping OpenBSD Secure Shell server...
Dec 26 15:38:48 ubuntu systemd[1]: ssh.service: Succeeded
Dec 26 15:38:48 ubuntu systemd[1]: Stopped OpenBSD Secure Shell server.
Lines 1-10 (END)
```

(a) pausa en Ubuntu

Timestamp	Name	Value
2021-12-26 16:39:09	SSH service is running	Down (0)
2021-12-26 16:38:09	SSH service is running	Up (1)
2021-12-26 16:37:09	SSH service is running	Up (1)
2021-12-26 16:36:09	SSH service is running	Up (1)
2021-12-26 16:33:09	SSH service is running	Up (1)
2021-12-26 16:32:09	SSH service is running	Up (1)
2021-12-26 16:31:09	SSH service is running	Up (1)
2021-12-26 16:30:09	SSH service is running	Up (1)
2021-12-26 16:29:09	SSH service is running	Up (1)

(b) Zabbix

Hacemos lo mismo en CentOS

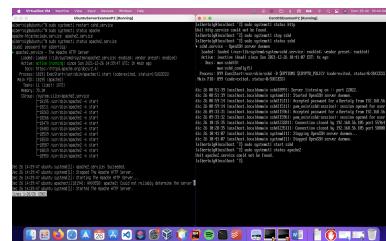
```
alberto@localhost:~$ sudo systemctl stop sshd
alberto@localhost:~$ sudo systemctl status sshd
● sshd.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Sun Dec 26 16:41:07 EST; 4s ago
     Docs: man:sshd(8)
           man:systemctl(1)
             Process: 899 ExecStart=/usr/sbin/sshd -D $OPTIONS _CRYPTD,_MULC (code=exited, status=0/SUCCESS)
             Main PID: 899 (code=exited, status=0/SUCCESS)
Dec 26 00:51:39 localhost.localdomain sshd[899]: Server listening on :: port 22.
Dec 26 00:51:39 localhost.localdomain sshd[111]: Started OpenSSH server daemon.
Dec 26 00:51:14 localhost.localdomain sshd[1511]: Accepted password for alberto from 192.168.56.1
Dec 26 00:51:14 localhost.localdomain sshd[1511]: pam_unix(sshd:session): session opened for user al
Dec 26 00:51:14 localhost.localdomain sshd[1511]: pam_unix(sshd:session): session closed for user al
Dec 26 00:51:33:32 localhost.localdomain sshd[1596]: pam_unix(sshd:session): session opened for user al
Dec 26 00:51:33:32 localhost.localdomain sshd[1596]: pam_unix(sshd:session): session closed for user al
Dec 26 00:51:35 localhost.localdomain sshd[2281]: Connection closed by 192.168.56.185 port 5596 (0)
Dec 26 00:51:39 localhost.localdomain sshd[2281]: Connection closed by 192.168.56.185 port 5086 (0)
Dec 26 00:51:47 localhost.localdomain sshd[2303]: pam_unix(sshd:session): session opened for user al
Dec 26 00:51:47 localhost.localdomain sshd[2303]: pam_unix(sshd:session): session closed for user al
Dec 26 00:51:49 localhost.localdomain sshd[111]: Stopped OpenSSH server daemon.
Lines 1-18 (END)
```

(c) pausa en CentOS

Timestamp	Name	Value
2021-12-26 16:41:35	SSH service is running	Down (0)
2021-12-26 16:40:35	SSH service is running	Up (1)
2021-12-26 16:39:35	SSH service is running	Up (1)
2021-12-26 16:38:35	SSH service is running	Up (1)
2021-12-26 16:37:35	SSH service is running	Up (1)
2021-12-26 16:36:35	SSH service is running	Up (1)
2021-12-26 16:35:35	SSH service is running	Up (1)
2021-12-26 16:34:35	SSH service is running	Up (1)
2021-12-26 16:33:35	SSH service is running	Up (1)
2021-12-26 16:32:35	SSH service is running	Up (1)
2021-12-26 16:31:35	SSH service is running	Up (1)
2021-12-26 16:30:35	SSH service is running	Up (1)
2021-12-26 16:29:35	SSH service is running	Up (1)

(d) Zabbix

Comprobamos ahora HTTP pausándolo en Ubuntu Server. Como en CentOS no está instalado (ya que realicé la pila LAMP en otra máquina virtual) es lógico que nos aparezca en Zabbix que HTTP está caído.



(e) máquinas virtuales

CentOS7: HTTP service is running		
Timestamp	Name	Value
2021-12-26 16:43:34	HTTP service is running	Down (0)
2021-12-26 16:43:34	HTTP service is running	Down (0)
2021-12-26 16:43:34	HTTP service is running	Down (0)
2021-12-26 16:40:34	HTTP service is running	Down (0)
2021-12-26 16:40:34	HTTP service is running	Down (0)
UbuntuServer: HTTP service is running		
Timestamp	Name	Value
2021-12-26 16:44:08	HTTP service is running	Up (1)
2021-12-26 16:43:08	HTTP service is running	Up (1)
2021-12-26 16:42:08	HTTP service is running	Up (1)
2021-12-26 16:41:08	HTTP service is running	Up (1)
2021-12-26 16:40:08	HTTP service is running	Up (1)

(f) Zabbix

Por lo que comprobamos que nuestro servidor Zabbix funciona y monitoriza correctamente ambas máquinas.

2. Ejercicio 2

Usted deberá saber cómo instalar y configurar Ansible para poder hacer un ping a las máquinas virtuales de los servidores y ejecutar un comando básico (p.ej. el script de monitorización del RAID1). También debe ser consciente de la posibilidad de escribir acciones más complejas mediante playbooks escritos con YAML. Incluya capturas de pantalla del proceso con una breve descripción en el mismo documento que suba para el ejercicio de Zabbix.

2.1. Instalación y configuración de Ansible en Ubuntu Server

Podemos descargar Ansible desde los paquetes de Linux

```
$ sudo apt install ansible
```

Comprobamos que se ha instalado correctamente

```
albertolg@ubuntu:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/albertolg/.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 = 3.8.2 (default, Jul 16 2020, 14:00:26) [GCC 9.3.0]
albertolg@ubuntu:~$
```

Vemos ahora el inventario, que son la lista de ordenadores con los que vamos a trabajar. Podemos encontrarlo en **/etc/ansible/hosts**. Lo abrimos y modificamos añadiendo el host de CentOS para poder conectarnos.

```
#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10
centos ansible_port=22022 ansible_host=192.168.56.110 ansible_user=albertolg
```

Modificamos ahora el archivo de configuración interna de Ansible que podemos encontrar en **/etc/ansible/ansible.cfg**. Ahí descomentamos inventory, cambiamos remote_port a 22022 y descomentamos ssh_args (parametros para conectarnos a CentOS).

```

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

```

Para que Ansible funcione de la forma que queremos tenemos que configurar la conexión SSH por clave pública como hicimos en la práctica 2 de la asignatura. En la siguiente imagen podemos ver los pasos (no entrará en detalles ya que se vió en la práctica anterior)

```

alberto1g@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alberto1g/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alberto1g/.ssh/id_rsa
Your public key has been saved in /home/alberto1g/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:DvLFPCXULV8zfU0E0DZYCEJukHK+I6ZYMjnrogevtbY alberto1g@ubuntu
The key's randomart image is:
+---[RSA 3072]---+
|O . -+ +|
|, , . o + o |
|o o o + o |
|o o o . . |
|+ . . . S |
|+ . . + o |
|**o . . . |
|*#E- . . o |
|X:E . . . |
+---[SHA256]---+
alberto1g@ubuntu:~$ ssh-copy-id 192.168.56.110 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/alberto1g/.ssh/id_rsa.pub"
/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 all the new keys
alberto1g@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.

alberto1g@ubuntu:"$ "

```

Ya estaría Ansible configurado correctamente, para ello hacemos un ping a CentOS

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

Como vemos funciona correctamente. Probamos a ejecutar un comando básico como puede ser `lsblk` demostrando así que está ejecutando el comando en nuestro servidor CentOS.

The screenshot shows two terminal windows side-by-side. The left window, titled 'UbuntuServerExamenP2 [Running]', shows the command `ansible -m shell -a 'lsblk' centos` being run. The output indicates a 'CHANGED' status with `rc=0`. The right window, titled 'CentOSExamenP2 [Running]', shows the output of the `lsblk` command on a CentOS system. It lists various disk partitions and their details, such as NAME, MAJ:MIN, RM, SIZE, RO, TYPE, and MOUNTPOINT. The output includes entries for sda, sda1, sda2, sdb, sdb1, sdb2, and sr0, along with swap and root partitions.

Como nos dice el enunciado, Ansible también permite la realización de **playbooks** que nos permiten ejecutar varios comandos similar a un script. Ejecutaríamos un playbook de la siguiente manera

```
$ ansible-playbook playbook.yml
```

Referencias

[Ansible 2021] Ansible (2021). Manual de ansible. https://docs.ansible.com/ansible/latest/user_guide/index.html. [Online; consultada 27-Diciembre-2021].

[Zabbix SIA, 2021] Zabbix SIA (2021). Manual de zabbix. <https://www.zabbix.com/documentation/5.0/manual>. [Online; consultada 27-Diciembre-2021].