Práctica 3

Monitorización, Automatización y "Profiling"

Ingeniería de Servidores
Curso 2022/23

Carmen García Romero

ETSIIT
Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación

Índice

Introducción

En esta práctica vamos a trabajar con Zabbix, un sistema de monitorización de redes, y se documentarán los distintos pasos necesarios hasta completar la práctica, abarcando desde su instalación y configuración, hasta la monitorización de la máquina virtual.

En Zabbix se necesita un servidor para albergar el sistema de monitorización y un agente que permite la comunicación entre el sistema que se quiere monitorizar y el servidor. Se va a instalar el servidor en Ubuntu Server; el agente se va a instalar en RockyCentOS y en Ubuntu Server.

1.Instalación de Zabbix en Ubuntu Server

Lo primero que hacemos es descargarnos el repositorio de Zabbix, versión 5.0, y la distribución Focal para Ubuntu.

Descomprimimos el paquete, usando el comando **dpkg** con la opción **-i** para instalar el paquete en el directorio de paquetes e instalamos Zabbix Server en Ubuntu.

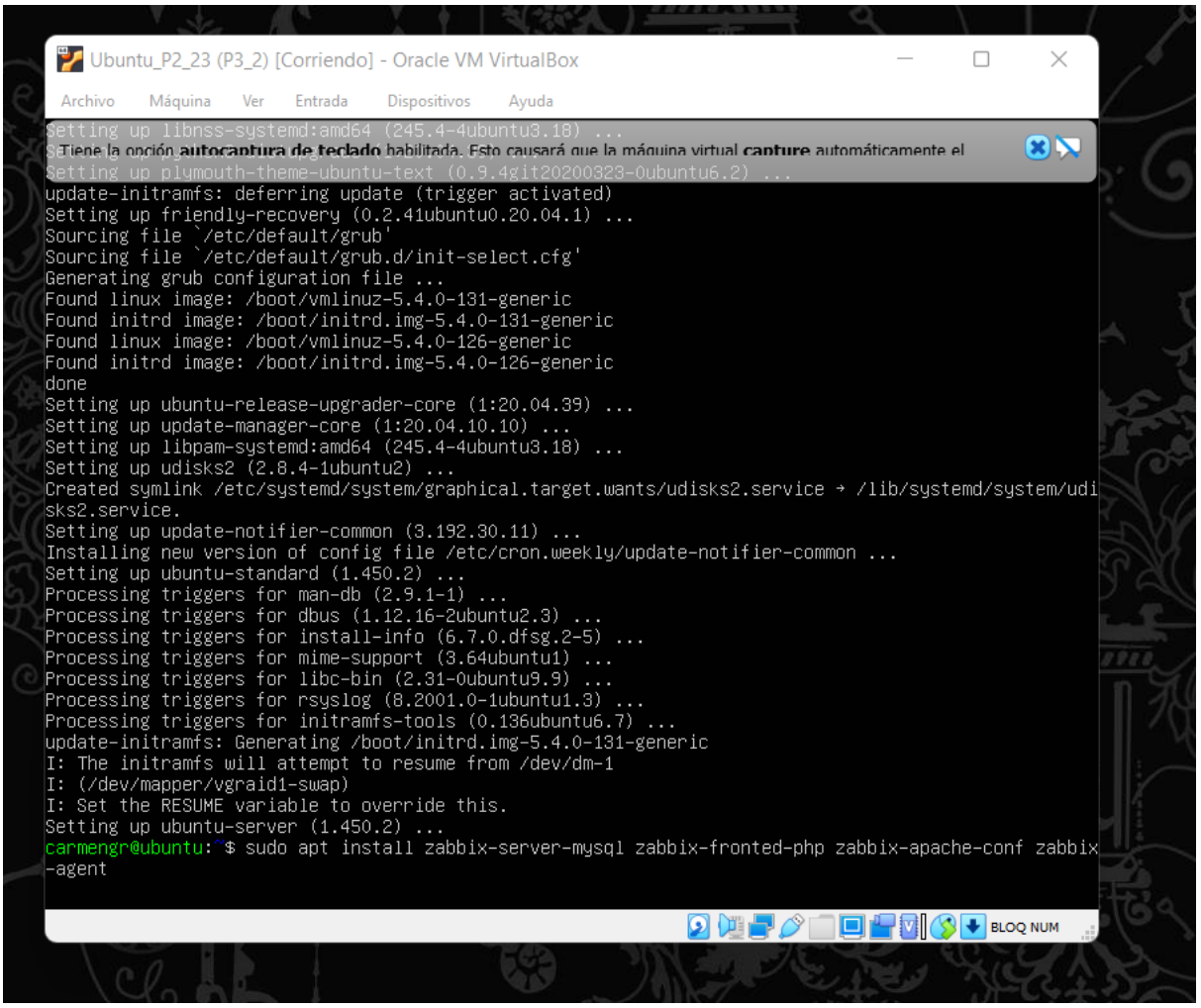
```
carmengr@ubuntu:~$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-
release_5.0-2%2Bubuntu22.04_all.deb
--2022-11-12 17:34:53--  https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix
-release_5.0-2%2Bubuntu22.04_all.deb
Resolving repo.zabbix.com (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: 4296 (4.2K) [application/octet-stream]
Saving to: ‘zabbix-release_5.0-2+ubuntu22.04_all.deb’

zabbix-release_5.0-2+ubu 100%[=================================>]   4.20K  --.-KB/s    in 0s

2022-11-12 17:34:54 (1.29 GB/s) - ‘zabbix-release_5.0-2+ubuntu22.04_all.deb’ saved [4296/4296]

carmengr@ubuntu:~$ dpkg -i zabbix-release_5.0-2+ubuntu22.04_all.deb
dpkg: error: requested operation requires superuser privilege
carmengr@ubuntu:~$ sudo dpkg -i zabbix-release_5.0-2+ubuntu22.04_all.deb
[sudo] password for carmengr:
Selecting previously unselected package zabbix-release.
(Reading database ... 71743 files and directories currently installed.)
Preparing to unpack zabbix-release_5.0-2+ubuntu22.04_all.deb ...
Unpacking zabbix-release (1:5.0-2+ubuntu22.04) ...
Setting up zabbix-release (1:5.0-2+ubuntu22.04) ...
carmengr@ubuntu:~$
```

Por último actualizamos la lista de paquetes con el comando: **sudo apt update.**

Instalación de servicios

Instalamos los servicios de zabbix necesarios para la práctica:

- **zabbix-server-mysql:** para el soporte de MySQL
- **zabbix-frontend-php**: el frontend
- **zabbix-apache-conf**
- **zabbix-agent:** para poder monitorizar el servidor



Nota**: Corrección de errata → zabbix-frontend-php***

Configuración de la base de datos Zabbix Server

Entramos a la base de datos MySQL como root y creamos la base de datos para Zabbix (zabbix), además creamos un usuario (zabbix), le damos una contraseña(zabbix) y le concedemos todos los permisos.

```
carmengr@ubuntu:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26
Server version: 8.0.31-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2022, 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 (0.01 sec)

mysql> create user zabbix@localhost identified by 'zabbix';
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;
```

Nota: Cabe recordar que usar el mismo nombre para todo es una mala práctica, pero para facilitar el desarrollo de esta práctica se ha hecho así.

Importamos el esquema y los datos iniciales

```
carmengr@ubuntu:~$ sudo zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -u zabbix -p
zabbix
Enter password:
```

Configurar la base de datos para el servidor Zabbix

Para ello editamos el fichero /etc/zabbix/zabbix_server.conf, configuramos el usuario y su contraseña.

```
### 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=zabbix

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

### Option: DBPort
#       Database port when not using local socket.
#
# Mandatory: no
-- INSERT --                                                    124,18        11%
```

Configurar PHP para la interfaz de Zabbix

Configuramos PH para zabbix frontend en el fichero /etc/zabbix/apache.conf; solo tenemos que descomentar la zona horaria de Europa, y cambiando la opción que viene por defecto por Madrid.

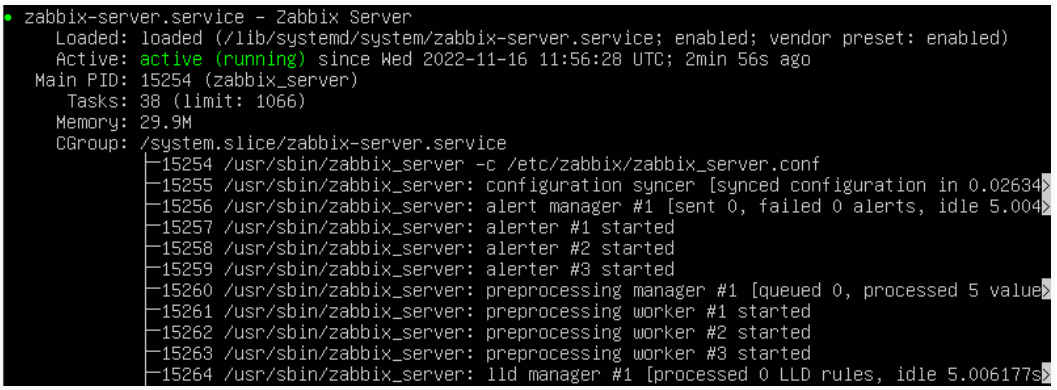
```
# Define /zabbix alias, this is the default
<IfModule mod_alias.c>
    Alias /zabbix /usr/share/zabbix
</IfModule>

<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
"/etc/zabbix/apache.conf" 77L, 1835C                          30,45     Comienzo
```

Reiniciamos y habilitamos los servicios

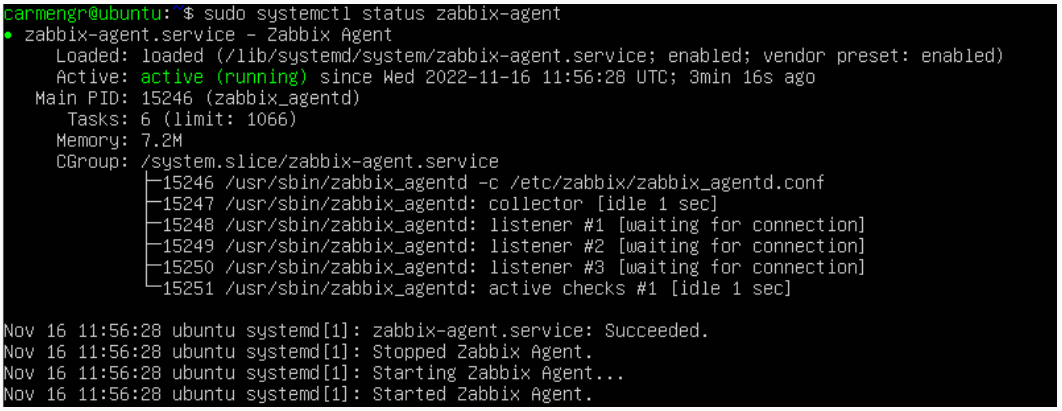
```
carmengr@ubuntu:~$ sudo systemctl restart zabbix-server zabbix-agent apache2
carmengr@ubuntu:~$ sudo systemctl enabe zabbix-server zabbix-agent apache2
Unknown operation enabe.
carmengr@ubuntu:~$ sudo systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service 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-insta
ll.
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.
carmengr@ubuntu:~$
```

Comprobamos sus estados

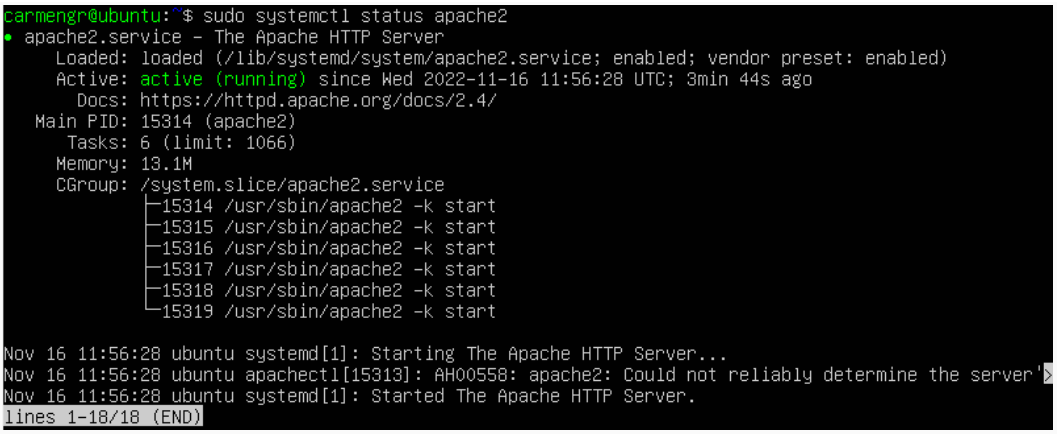
- Server



- Agent

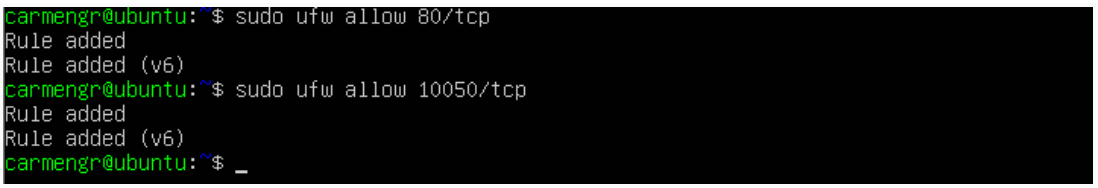


- Apache



Habilitamos el puerto

Habilitamos los puertos 80 y 10050, siendo este último el que usa Zabbix por defecto.



Configuramos del agente

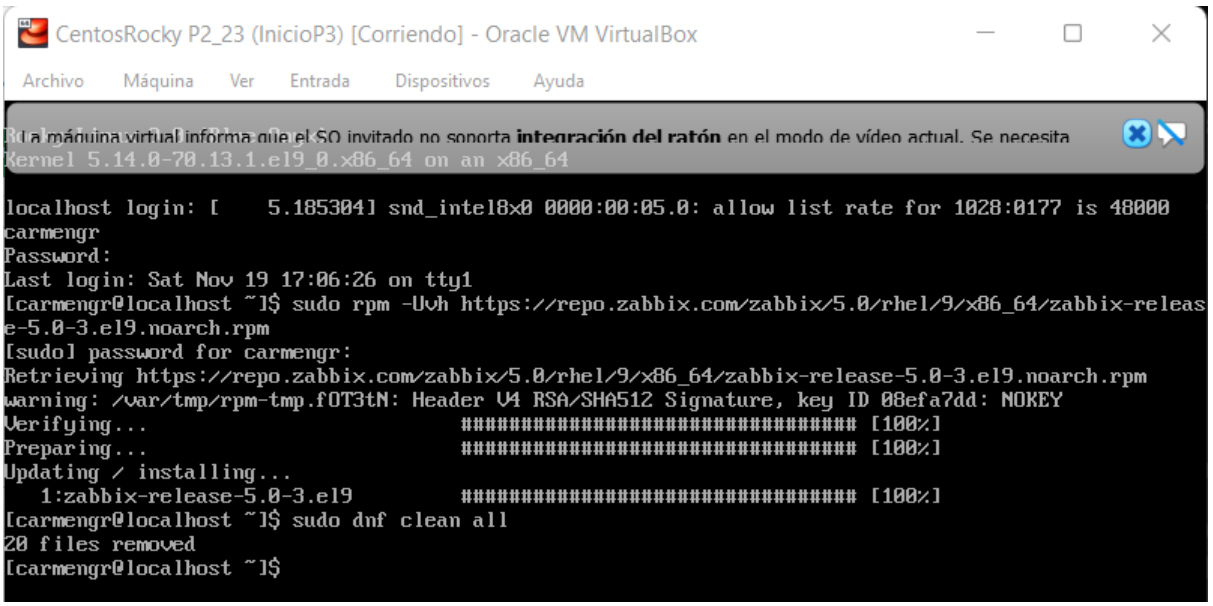
Accediendo al archivo /etc/zabbix/zabbix_agentd.conf, aqui indicamos la ip del servidor, por defecto tendrá la del localhost, modificamos **Server** y **ServerActive**; esto nos permitirá monitorizar fuera del servidor Ubuntu.

```
### Option: Server
#       List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix se
rvers 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 e
qually
#       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
```

```
#       If this parameter is not specified, active checks are disabled.
#       Example: ServerActive=127.0.0.1:20051,zabbix.example.com,[::1]:30051,::1,[12fc::1]
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.56.105

### 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=
```

2.Instalación y configuración del agente. CentosRocky

Descargamos el paquete zabbix del repositorio



Instalamos solo el agente Zabbix

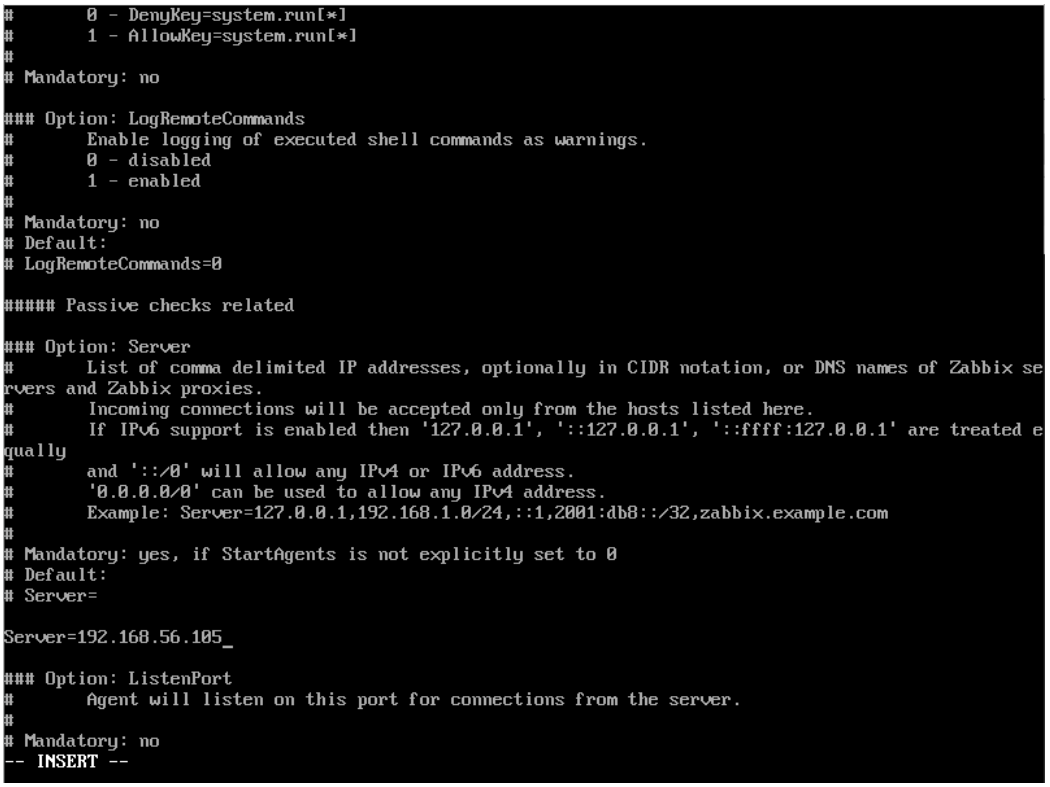
```
  Verifying        : libsss_nss_idmap-2.6.2-4.el9_0.x86_64                   15/23
  Verifying        : libsss_idmap-2.6.2-4.el9_0.1.x86_64                     16/23
  Verifying        : libsss_idmap-2.6.2-4.el9_0.x86_64                       17/23
  Verifying        : libsss_certmap-2.6.2-4.el9_0.1.x86_64                   18/23
  Verifying        : libsss_certmap-2.6.2-4.el9_0.x86_64                     19/23
  Verifying        : libcurl-7.76.1-14.el9_0.5.x86_64                        20/23
  Verifying        : libcurl-7.76.1-14.el9.x86_64                            21/23
  Verifying        : openldap-2.6.2-1.el9_0.x86_64                           22/23
  Verifying        : openldap-2.4.59-4.el9_0.x86_64                          23/23

Upgraded:
  gnupg2-2.3.3-2.el9_0.x86_64                      libcurl-7.76.1-14.el9_0.5.x86_64
  libsss_certmap-2.6.2-4.el9_0.1.x86_64            libsss_idmap-2.6.2-4.el9_0.1.x86_64
  libsss_nss_idmap-2.6.2-4.el9_0.1.x86_64          libsss_sudo-2.6.2-4.el9_0.1.x86_64
  openldap-2.6.2-1.el9_0.x86_64                    sssd-client-2.6.2-4.el9_0.1.x86_64
  sssd-common-2.6.2-4.el9_0.1.x86_64               sssd-kcm-2.6.2-4.el9_0.1.x86_64
Installed:
  libevent-2.1.12-6.el9.x86_64                 openldap-compat-2.6.2-1.el9_0.x86_64
  zabbix-agent-5.0.29-1.el9.x86_64

Complete!
[carmengr@localhost ~]$ sudo dnf install zabbix-agent
```

Iniciamos el agente de Zabbix

```
[carmengr@localhost ~]$ sudo systemctl enable zabbix-agent.service
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /usr/lib/systemd/
system/zabbix-agent.service.
[  423.404029] systemd-rc-local-generator[1821]: /etc/rc.d/rc.local is not marked executable, skippi
ng.
[carmengr@localhost ~]$ sudo systemctl status zabbix-agent.service
● zabbix-agent.service - Zabbix Agent
     Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: disabled)
     Active: inactive (dead)
[carmengr@localhost ~]$ sudo systemctl start zabbix-agent.service
[carmengr@localhost ~]$ sudo systemctl status zabbix-agent.service
● zabbix-agent.service - Zabbix Agent
     Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: disabled)
     Active: active (running) since Sat 2022-11-19 17:21:18 CET; 1s ago
    Process: 1839 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
   Main PID: 1841 (zabbix_agentd)
      Tasks: 6 (limit: 5896)
     Memory: 4.8M
        CPU: 13ms
     CGroup: /system.slice/zabbix-agent.service
             ├─1841 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
             ├─1842 "/usr/sbin/zabbix_agentd: collector [idle 1 sec]"
             ├─1843 "/usr/sbin/zabbix_agentd: listener #1 [waiting for connection]"
             ├─1844 "/usr/sbin/zabbix_agentd: listener #2 [waiting for connection]"
             ├─1845 "/usr/sbin/zabbix_agentd: listener #3 [waiting for connection]"
             └─1846 "/usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]"

Nov 19 17:21:18 localhost.localdomain systemd[1]: Starting Zabbix Agent...
Nov 19 17:21:18 localhost.localdomain systemd[1]: Started Zabbix Agent.
[carmengr@localhost ~]$
```

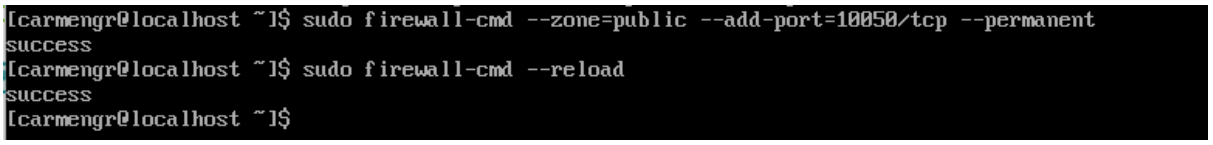
Configuración del agente

Configuramos el agente accediendo al archivos **/etc/zabbix/zabbix_agentd.conf.** Añadimos la ip del servidor de Ubuntu, tanto en **Server(que tendrá la IP local por defecto)** como en **ServerActive.** De esta forma podremos monitorizar el servidor desde una máquina externa, en este caso, CentosRocky.



Habilitamos el puerto

Habilitamos el puerto **10050,** que es el que usa por defecto el agente de Zabbix. Para ello usamos dos comandos, ya que uno abre el puerto de forma inmediata pero temporal, el otro lo dejará abierto al reiniciar la máquina.

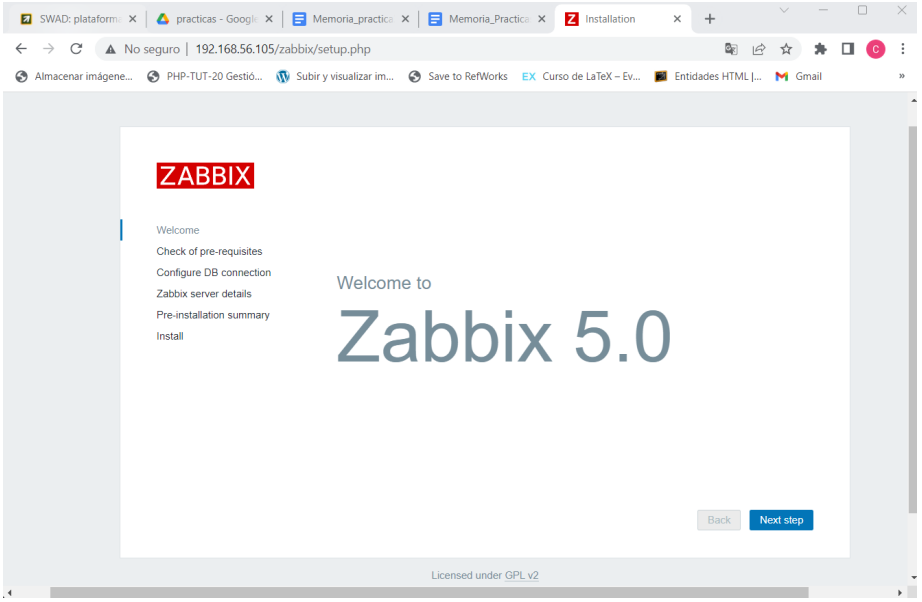


Hecho esto, reiniciamos los servicios y ya podremos empezar a monitorizar.

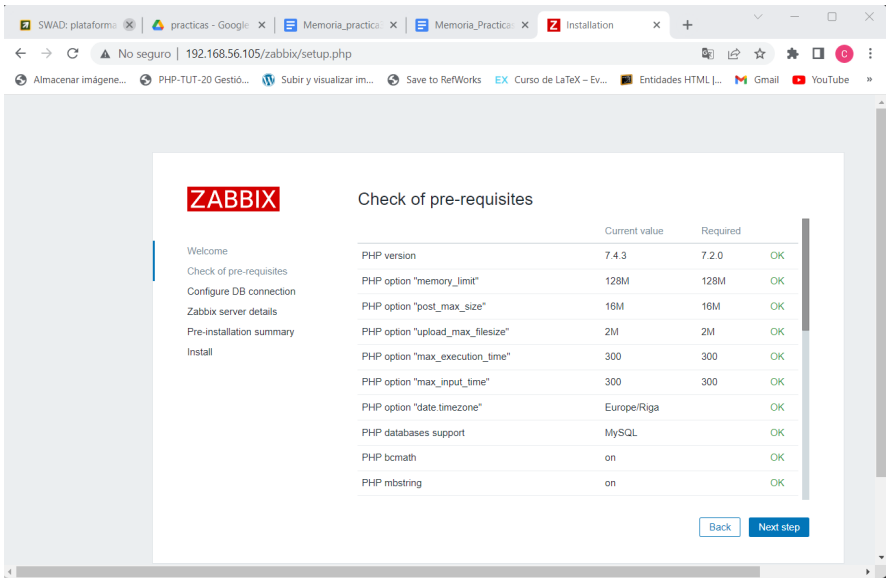
3. FontEnd

3.1 Configuración del servidor desde el navegador

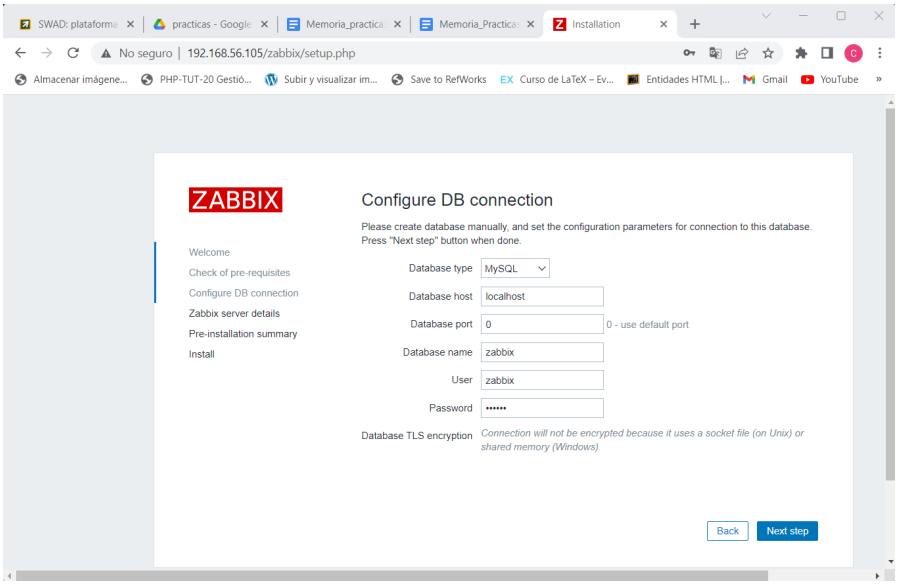
Entramos en la dirección http://192.168.56.105 para acceder a Zabbix server y configurarlo.

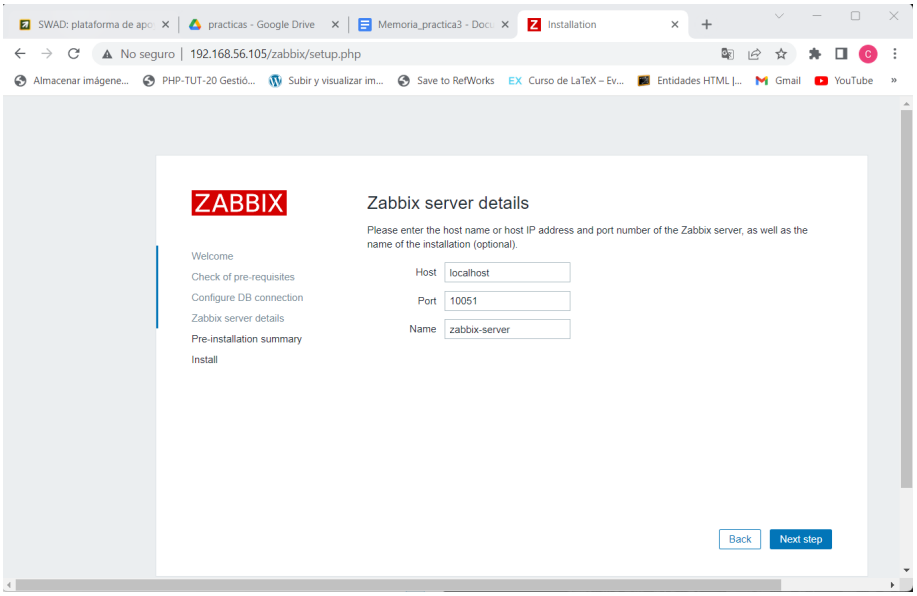


Comprobamos que lo instalado está bien configurado.

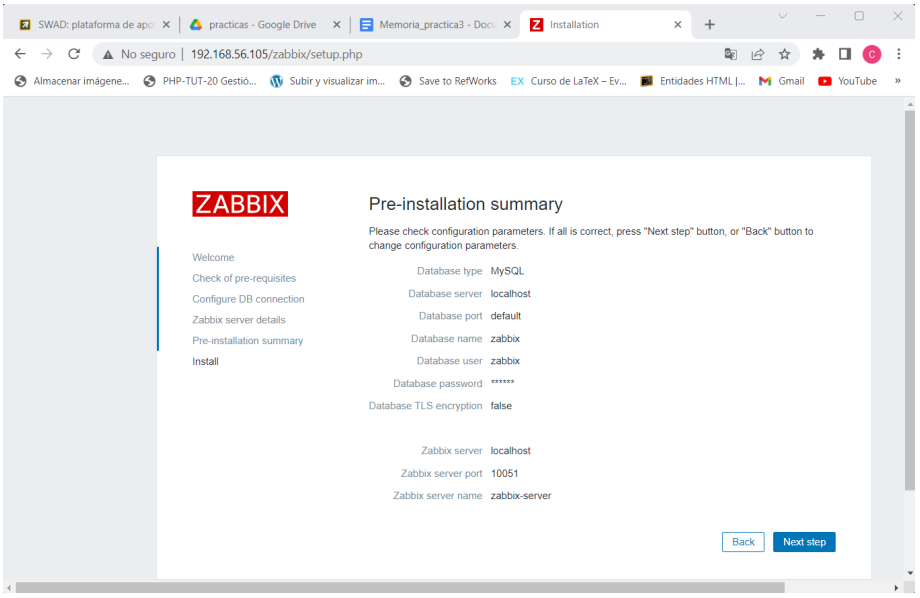


Conexión con la base de datos





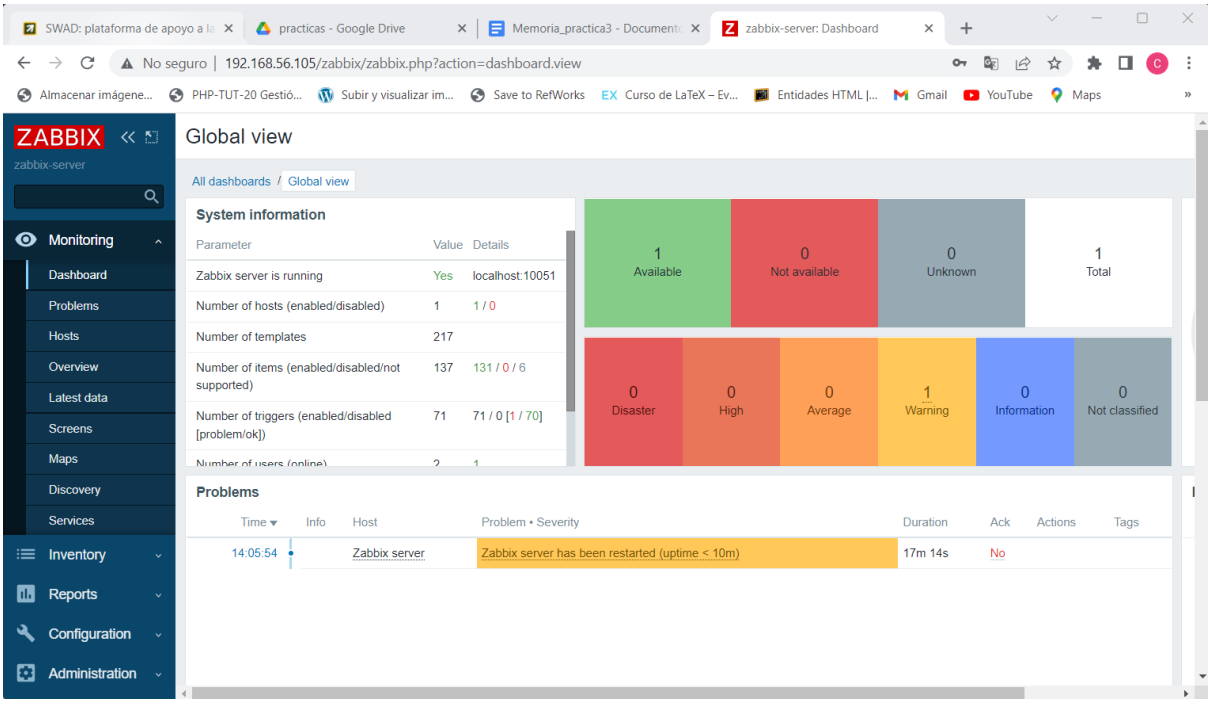
Comprobamos que los datos sean correctos



Tras terminar la configuración, veremos una ventana que nos confirma que se ha instalado correctamente el Frontend y ya podremos acceder.
Hacemos el Log in, datos:

- user : Admin
- password: zabbix

Y accedemos al dashboard principal

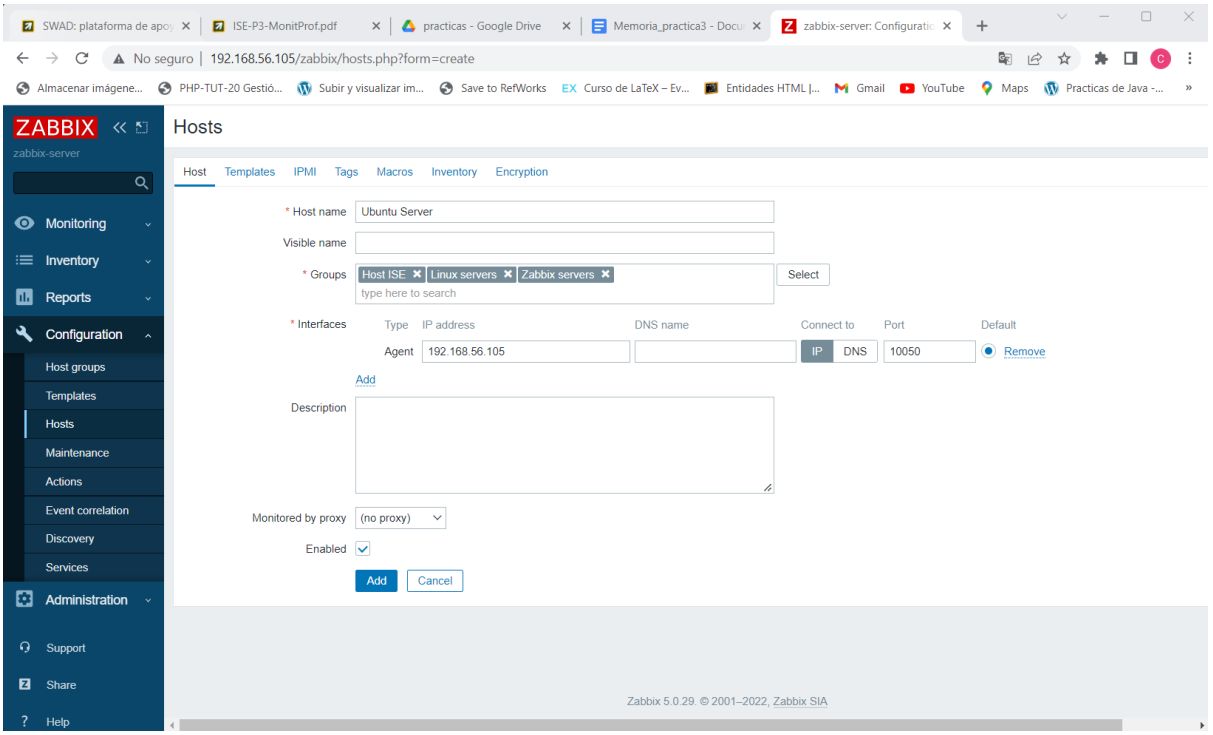


3.2 Configuración de la monitorización de los servicios SSH y HTTP

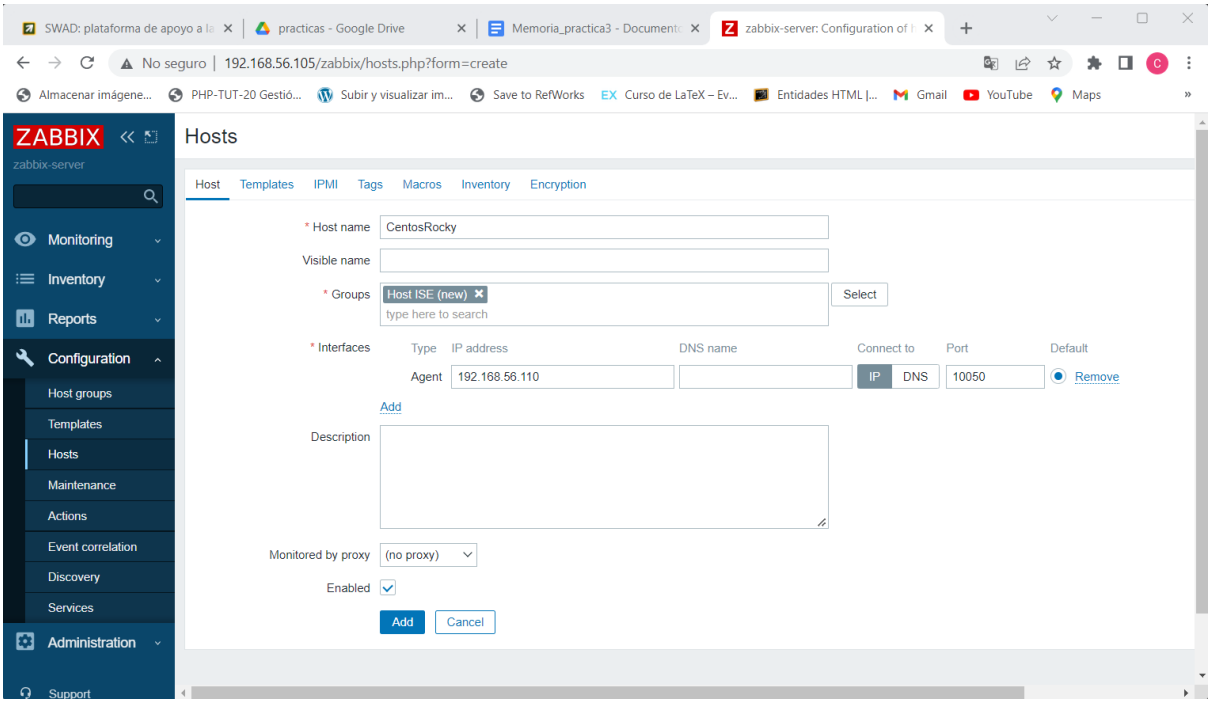
Creación de Hosts

En Configuration → Host → new Host, creamos un Host para el servidor de Ubuntu y otro para el agente de CentosRocky.

Servidor Ubuntu

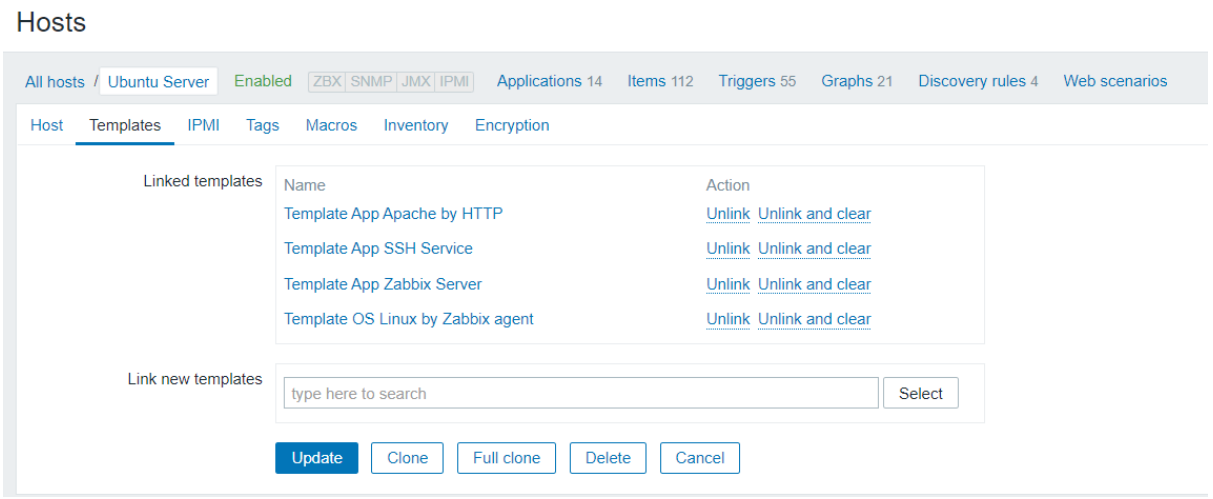


Agente de CentosRocky

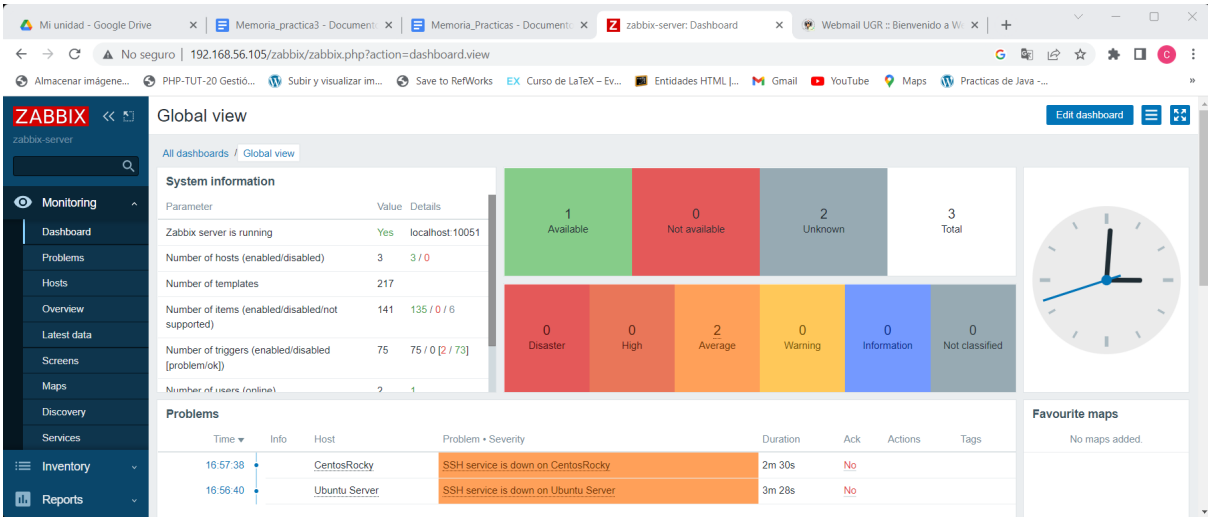


Creación de los templates

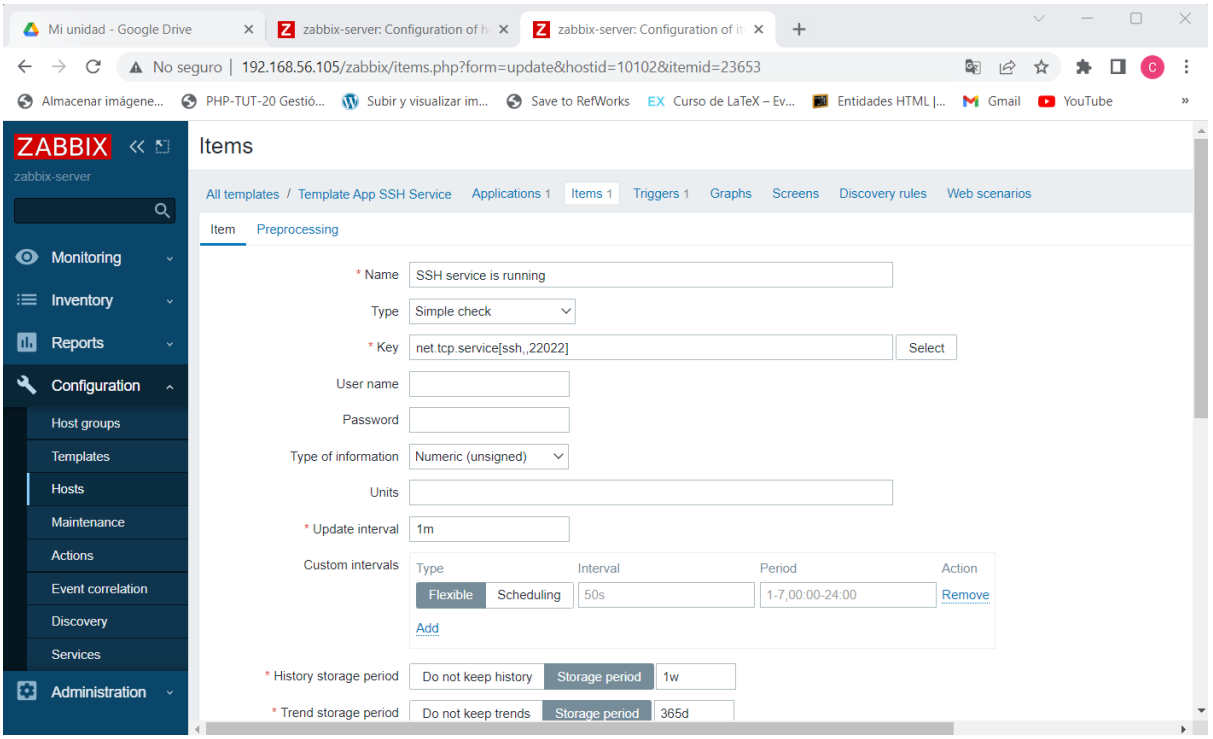
Los templates nos servirán para monitorizar los servicios del servidor, en nuestro caso SSH y HTTP. Para ello, modificamos los hosts creados, nos vamos a templates y añadimos las plantillas Template App SSH y Template App HTTP Service.



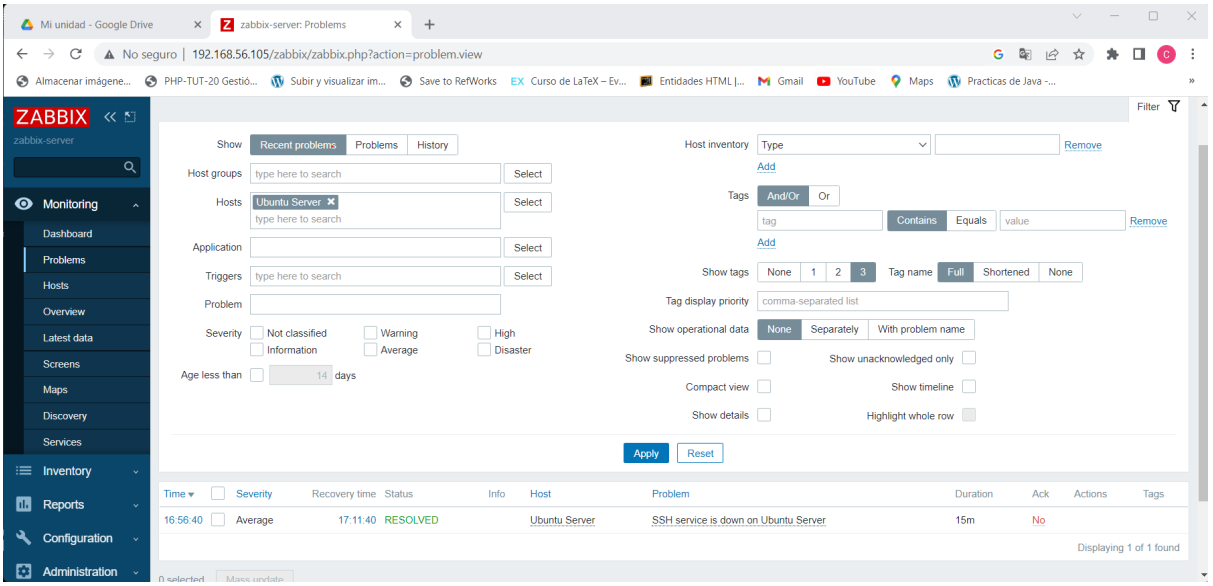
A continuación, el servicio ssh nos da error, pese a que en las máquinas virtuales está activo. Esto se debe a que en prácticas anteriores cambiamos el puerto 22 por 22022, así que tendremos que configurarlo, ya que el template SSH trae por defecto el puerto 22.



Nos vamos a la configuración del Template App SSH Service → Items 1 → SSH service is running y añadimos el puerto 22022

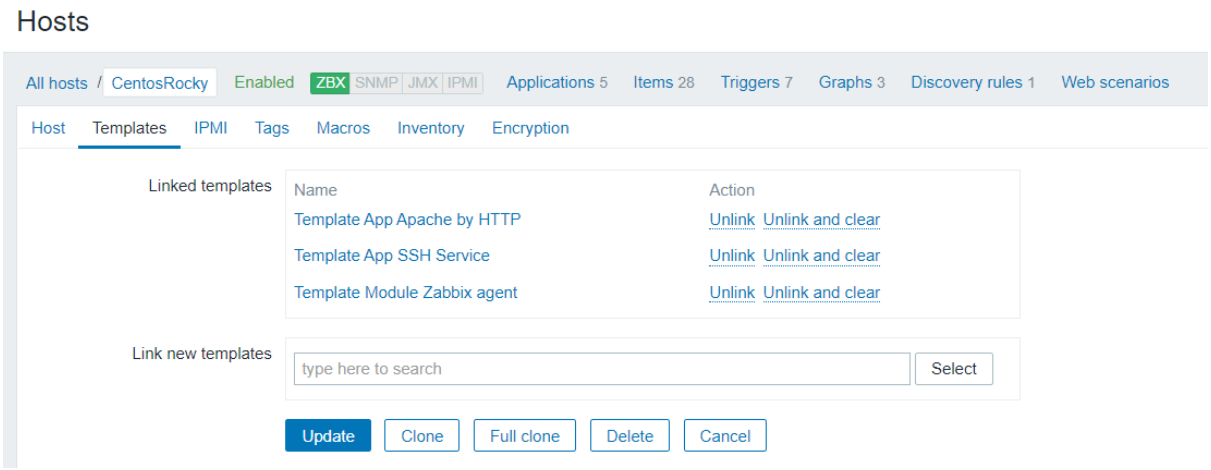


Una vez hecho esto veremos que el problema que nos daba anteriormente del servicio ssh aparece como resuelto.



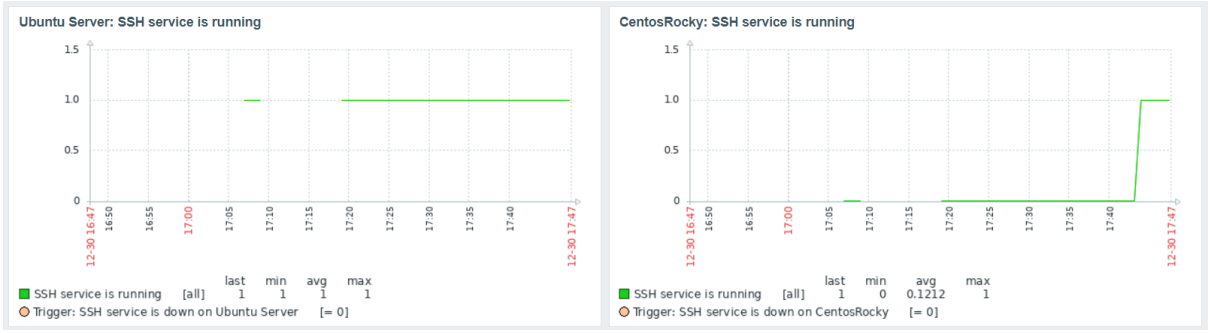
Añadimos los templates para CentosRocky

Esto nos servirá para poder monitorizar los servicios para el host CentosRocky.



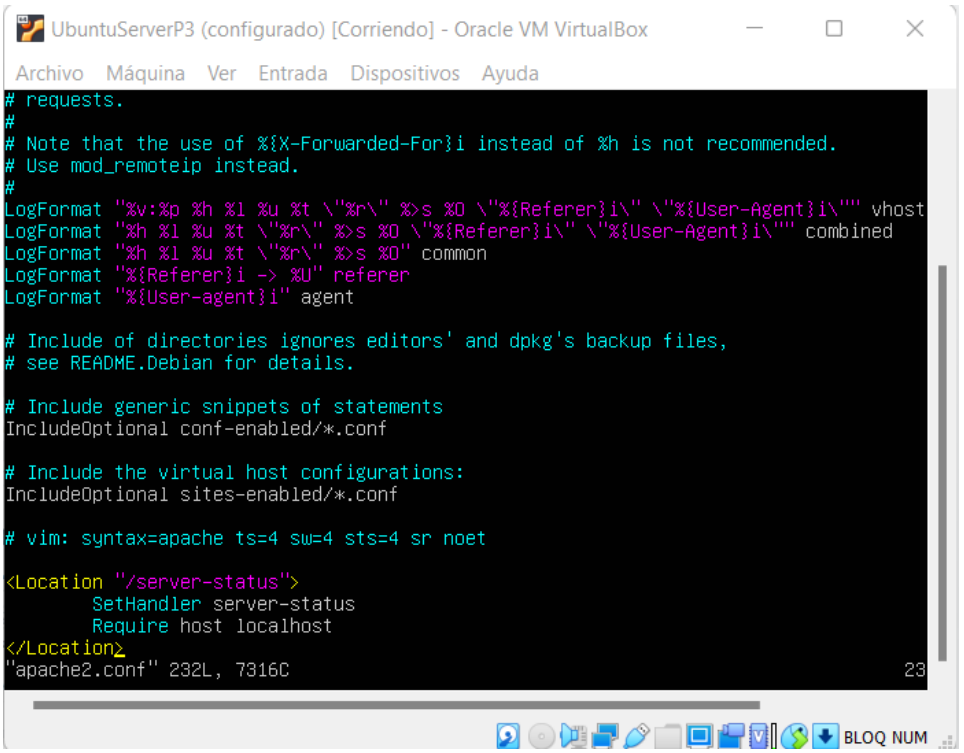
En este caso no será necesario modificar de nuevo el puerto, porque al modificar el Template SSH la modificación del puerto se aplica también al añadir el template a CentosRocky.

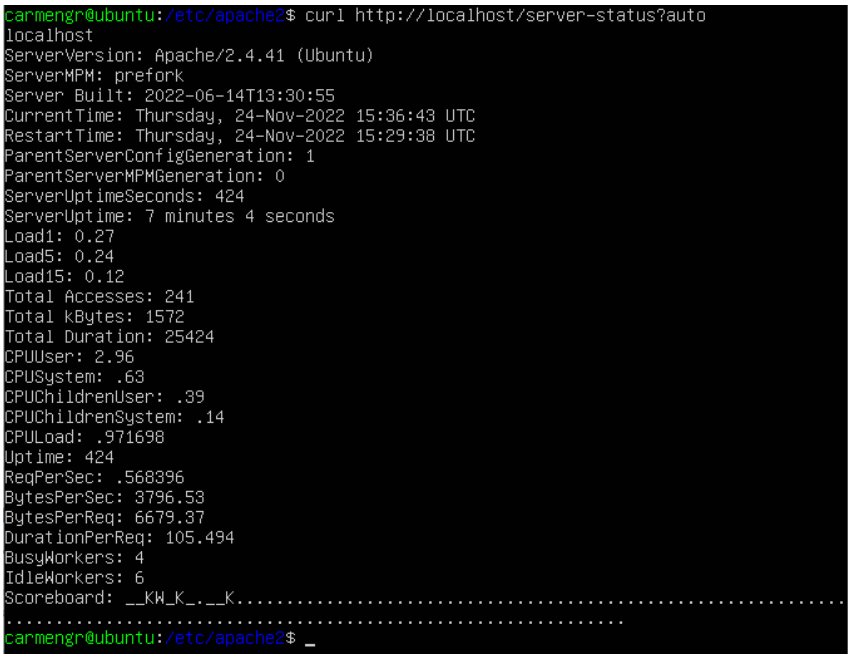
Añadimos widget para la monitorización



4.Problemas encontrados durante el desarrollo de la práctica

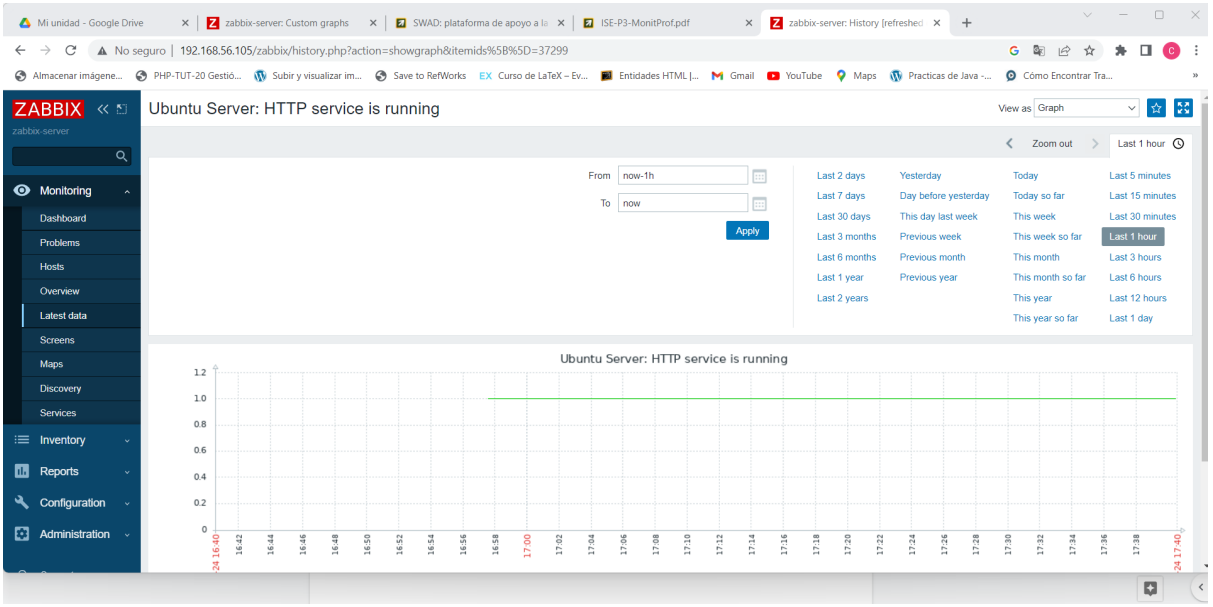
- Fallo en la monitorización
  Encuentro un problema, Zabbix no me llega a monitorizar los servicios, no me muestra gráficas en el monitoreo. Añado el siguiente código al fichero /etc/apache2/apache.config para que salga la página web con la información de apache, para monitorizarlo.





y comprobamos que está funcionando correctamente, por tanto el error está en la conexión con zabbix.

Nota1: No he conseguido monitorizar la información completa de apache en zabbix de manera gráfica, en Latest data, pero con la siguiente captura vemos que efectivamente está monitorizando el servicio correctamente.



Nota2: Tras un problema con VirtualBox, levanté de nuevo las máquinas virtuales e hice de nuevo toda la práctica. Me di cuenta que en la configuración del agente la primera vez que lo hice, me equivoqué escribiendo la IP del servidor, puede que ese fuera el motivo del error anterior.

Finalmente pude añadí unos widget para la monitorización de servicio ssh y ya si pude obtener la información correctamente.

Parte II : Ansible

Introducción

En esta segunda parte de la práctica vamos a instalar y configurar Ansible para poder hacer un ping a las máquinas virtuales de los servidores. Ansible se trata de una herramienta de automatización que se instala en un nodo de control el cual administra dispositivos de forma remota, por defecto sobre SSH.

1.Instalación y configuración de Ansible en Ubuntu Server

En primer lugar actualizamos los repositorios y a continuación instalamos la instalación de Ansible.

```
carmengr@ubuntu:~$ sudo apt-add-repository --yes --update ppa:ansible/ansible
Obj:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease
Obj:4 http://es.archive.ubuntu.com/ubuntu focal-security InRelease
Obj:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease
Des:6 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18,0 kB]
Des:7 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 Packages [1.132 B]
Des:8 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main Translation-en [756 B]
Descargados 19,9 kB en 6s (3.412 B/s)
Leyendo lista de paquetes... Hecho
carmengr@ubuntu:~$ sudo apt install ansible_
```

Comprobamos la versión instalada

```
carmengr@ubuntu:~$ ansible --version
ansible [core 2.12.10]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/carmengr/.ansible/plugins/modules', '/usr/share/ansible/p
ugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/carmengr/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.8.10 (default, Nov 14 2022, 12:59:47) [GCC 9.4.0]
  jinja version = 2.10.1
  libyaml = True
carmengr@ubuntu:~$ _
```

A continuación, vamos a la lista de máquinas a las que vamos a conectar con ansible, dicha lista se encuentra en /etc/ansible/host. Dentro añadiremos el nombre del módulo con el que llamaremos en ansible, el puerto, la IP de la máquina CentosRocky a la que se conectará, el usuario con el que se conectara y los parámetros ssh.

```
#   - 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

192.168.56.105
192.168.56.110

centosrocky ansible_host=192.168.56.110 ansible_user=carmengr ansible_port=22022
ubuntu ansible_host=192.168.56.105 ansible_user=carmengr  ansible_port=22022


# 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
-- INSERTAR --                                                    23,1      Comienzo
```

Ansible utiliza ssh por defecto, por tanto debemos tenerlo instalado en nuestras máquinas, crear la llave pública y privada en nuestra máquina anfitriona y enviar la llave pública a la máquina CentOS y Ubuntu. Para ello usamos el comando: **ssh-keygen**

```
carmengr@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/carmengr/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/carmengr/.ssh/id_rsa
Your public key has been saved in /home/carmengr/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:dhgo3ixVpVT1hh8HavUvsKOZwpzU7qmA6CwwtYaT2zw carmengr@ubuntu
The key's randomart image is:
+---[RSA 3072]----+
|        ooo.. o  |
|       + . . = o |
|    . o o . * + o|
|  .. =   * + * o.|
| + .o o S * + + .|
|* o  o o . o = . |
| O  . . .   =    |
|. Eo     . .     |
|   oo     .      |
+----[SHA256]-----+
carmengr@ubuntu:~$ _
```

Y copiamos la llave pública en las máquinas con el comando **ssh-copy-id <IP_maquina>**

```
carmengr@ubuntu:~$ ssh-copy-id 192.168.56.110 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/carmengr/.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 alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
carmengr@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.

carmengr@ubuntu:~$ _
```

```
carmengr@ubuntu:~$ ssh-copy-id 192.168.56.105 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/carmengr/.ssh/id_rsa.pub"
The authenticity of host '[192.168.56.105]:22022 ([192.168.56.105]:22022)' can't be established.
ECDSA key fingerprint is SHA256:DCWixfRbddSA6OefFYtjOGxWsWVg3+/7ywwvehRxwKM.
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 alr
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to inst
all the new keys
carmengr@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.

carmengr@ubuntu:~$
```

2.Ejecutamos ansible

Conexión

Tras copiar nuestra clave pública, comprobamos que tenemos conexión desde Ansible a las máquinas

```
carmengr@ubuntu:~$ ansible  -m ping centosrocky
[WARNING]: Platform linux on host centosrocky is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning
of that path. See https://docs.ansible.com/ansible-
core/2.12/reference_appendices/interpreter_discovery.html for more information.
centosrocky | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.9"
    },
    "changed": false,
    "ping": "pong"
}
carmengr@ubuntu:~$ ansible -m ping ubuntu
ubuntu | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
carmengr@ubuntu:~$ _
```

Ya podemos conectarnos a las máquinas usando Ansible. Además podemos mandar comandos a ambas máquinas simultáneamente.

```
carmengr@ubuntu:~$ ansible -m shell -a 'ls -la' all
```

- **all** : para lanzarlo en todas las máquinas
- **-m** shell
- **-a 'comando'** : para especificar el comando a ejecutar

```
ubuntu | CHANGED | rc=0 >>
total 44
drwxr-xr-x 5 carmengr carmengr 4096 dic 30 17:09 .
drwxr-xr-x 4 root     root     4096 dic 14 11:25 ..
drwxrwxr-x 4 carmengr carmengr 4096 dic 30 17:43 .ansible
-rw------- 1 carmengr carmengr 3366 dic 30 17:41 .bash_history
-rw-r--r-- 1 carmengr carmengr  220 feb 25  2020 .bash_logout
-rw-r--r-- 1 carmengr carmengr 3770 dic 30 16:57 .bashrc
drwx------ 2 carmengr carmengr 4096 dic 14 11:26 .cache
-rw-r--r-- 1 carmengr carmengr  807 feb 25  2020 .profile
drwx------ 2 carmengr carmengr 4096 dic 30 17:40 .ssh
-rw-r--r-- 1 carmengr carmengr    0 dic 14 11:26 .sudo_as_admin_successful
-rw-r--r-- 1 root     root     4244 may 28  2020 zabbix-release_5.0-1+focal_all.deb
[WARNING]: Platform linux on host centosrocky is using the discovered Python interpreter at
/usr/bin/python3.9, but future installation of another Python interpreter could change the meaning
of that path. See https://docs.ansible.com/ansible-
core/2.12/reference_appendices/interpreter_discovery.html for more information.
centosrocky | CHANGED | rc=0 >>
total 28
drwx------. 4 carmengr carmengr  153 Dec 30 18:50 .
drwxr-xr-x. 3 root     root       22 Dec 15 20:16 ..
drwx------. 3 carmengr carmengr   17 Dec 30 18:50 .ansible
-rw-------. 1 carmengr carmengr 3873 Dec 30 18:49 .bash_history
-rw-r--r--. 1 carmengr carmengr   18 May 16  2022 .bash_logout
-rw-r--r--. 1 carmengr carmengr  141 May 16  2022 .bash_profile
-rw-r--r--. 1 carmengr carmengr  492 May 16  2022 .bashrc
-rw-------. 1 carmengr carmengr   20 Dec 29 18:29 .lesshst
drwx------. 2 carmengr carmengr   71 Dec 30 18:37 .ssh
-rw-r--r--. 1 carmengr carmengr 1471 Dec 29 20:37 ZA.pp
-rw-r--r--. 1 carmengr carmengr  370 Dec 29 20:37 ZA.te
carmengr@ubuntu:~$
```

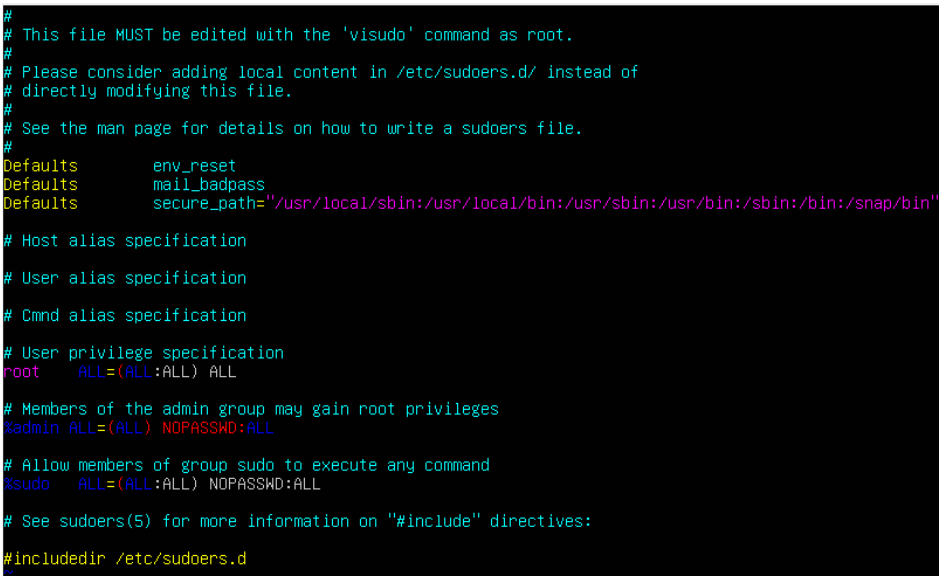
Ejecución del comando Poweroff

El objetivo de este apartado de la práctica es la ejecución del comando poweroff desde la maquina anfitriona, para que apague el resto de máquinas, simultáneamente. Para hacer esto tenemos que tener en cuenta que solo el superusuario puede ejecutar este comando. Por ello, tendremos que hacer una serie de modificaciones primero.

Modificar el archivo /etc/sudoers

El archivo sudoers incluye una lista de los usuarios que pueden usar el comando sudo para obtener privilegios de root. Modificaremos algunas líneas quedando de la siguiente forma:

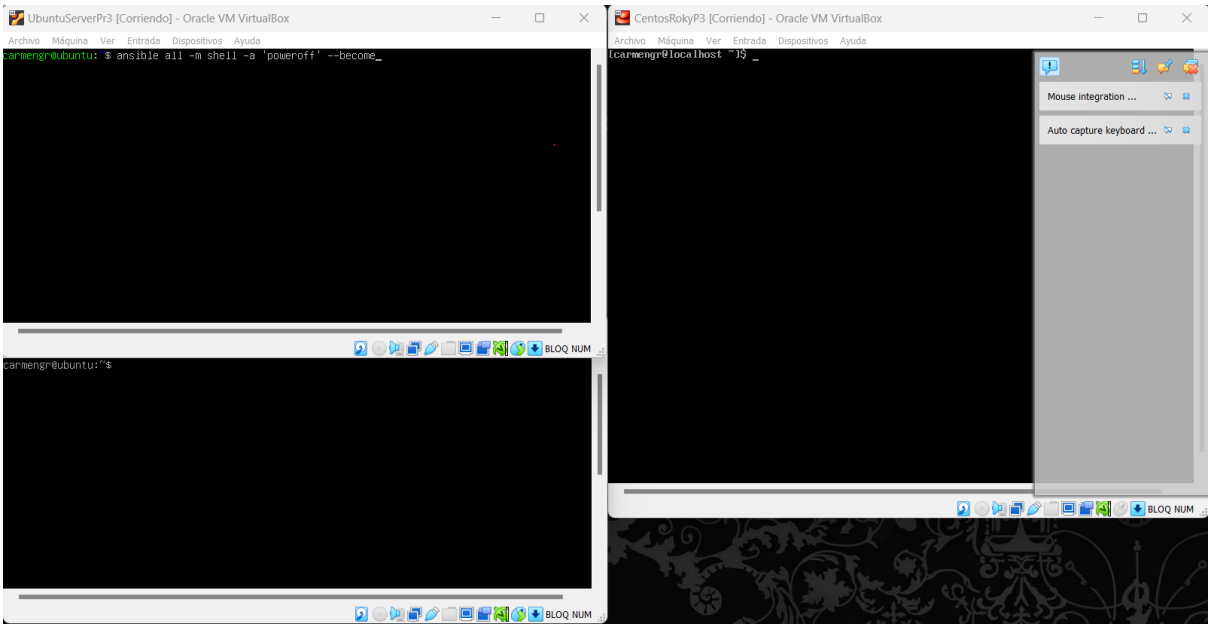
- admin ALL=(ALL) NOPASSWD:ALL
- sudo ALL = (ALL: ALL) NOPASSWD:ALL



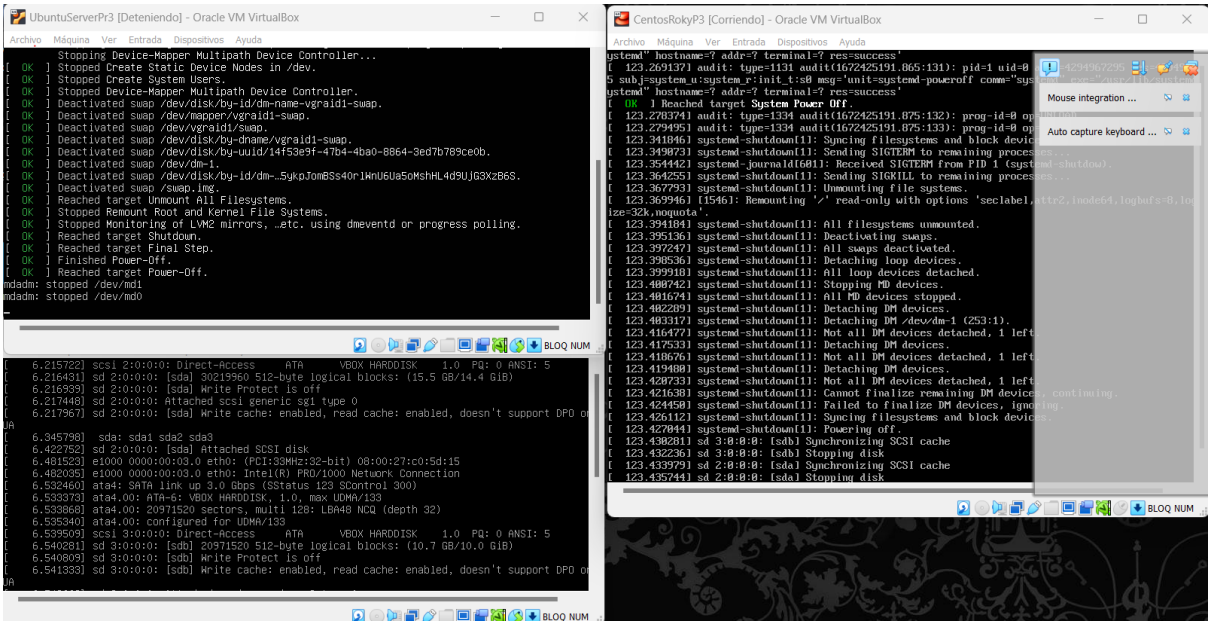
```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) NOPASSWD:ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) NOPASSWD:ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
```

Poweroff

A.



B.



Fin Práctica 3.

Referencias

https://www.zabbix.com/la/download?zabbix=5.0&os_distribution=alma_linux&os_version=9&components=agent&db=&ws=
https://stackoverflow.com/questions/68200965/error-2002-hy000-cant-connect-to-local-mysql-server-through-socket-var-run
https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-ansible-on-ubuntu-20-04-es