

ISE 22/23 - Práctica 3: Monitorización, Automatización y Profiling

Clara María Romero Lara



UNIVERSIDAD DE GRANADA

Ejercicio 1: Zabbix

Realice una instalación de Zabbix 5.0 en su servidor con Ubuntu Server 20.04 y configure para que se monitoree 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.

Instalación en Ubuntu Server

Partimos de la configuración previa de la P1L1, Ubuntu 20.04 con configuración de red y de ssh completa. Nos conectaremos a la máquina virtual desde nuestra máquina host mediante SSH, para mayor comodidad.

Lo primero será acceder a la [página oficial de Zabbix](#) y rellenar nuestras especificaciones. La página generará automáticamente las instrucciones para la instalación en el sistema.

- Instalar el repositorio:

```
> wget
https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-
-release/zabbix-release_5.0-1%2Bfocal_all.deb
> dpkg -i zabbix-release_5.0-1+focal_all.deb
> apt update
```

```
clararl@practicasis:~$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1%2Bfocal_all.deb
--2022-12-08 01:09:13-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1%2Bfocal_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+fo 100%[=====>] 4.14K --.-KB/s in 0s
2022-12-08 01:09:14 (575 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]
```

```
clararl@practicasis:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
[sudo] password for clararl:
Selecting previously unselected package zabbix-release.
(Reading database ... 71439 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) ...
clararl@practicasis:~$ apt update
```

- Instalar el server, frontend y agente:

```
> apt install zabbix-server-mysql zabbix-frontend-php
zabbix-apache-conf zabbix-agent
```

- Crear base de datos inicial

- Nuestra primera incidencia: tenemos que hacer un pequeño inciso para [instalar MySQL](#), ya que en la práctica 2 la pila LAMP se trabajó en CentOS.

```
> sudo apt install mysql-server
> sudo systemctl start mysql.service
> sudo mysql
>> ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'password';
>> exit
> mysql-secure-installation
> mysql -u root -p
>> ALTER USER 'root'@'localhost' IDENTIFIED WITH
auth_socket;
```

```
clararl@practicasis:~$ sudo systemctl start mysql.service
clararl@practicasis:~$ sudo systemctl status mysql.service
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-08 01:26:52 UTC; 7s ago
     Process: 16224 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 16242 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 1066)
      Memory: 363.7M
    CGroup: /system.slice/mysql.service
            └─16242 /usr/sbin/mysqld
```

- Una vez instalado, continuamos con la creación de la base de datos inicial.

```
> mysql -u root -p
>> create database zabbix character set utf8 collate
utf8_bin;
>> create user zabbix@localhost identified by 'password';
>> grant all privileges on zabbix.* to zabbix@localhost;
>> set global log_bin_trust_function_creators = 1;
>> quit;
```

```
made so far will take effect immediately.
Reload privilege tables now? (Press y|Y for Yes, any other key for No) :
... skipping.
All done!
clararl@practicasis:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 8.0.31-0ubuntu0.20.04.2 (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> ALTER USER 'root'@'localhost' IDENTIFIED WITH auth_socket;
Query OK, 0 rows affected (0.03 sec)

mysql> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected, 2 warnings (0.03 sec)

mysql> create user zabbix@localhost identified by 'password';
Query OK, 0 rows affected (0.06 sec)

mysql> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.04 sec)

mysql> set global log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- Importar el esquema inicial del server de Zabbix. Tardará un poco. Para comprobar si se ha importado correctamente, podemos volver a ejecutar el comando.

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

- Deshabilitamos `log_bin_trust_function_creators`

```
> sudo mysql
>> set global log_bin_trust_function_creators = 0;
>> quit;
```

- Configuramos la base de datos, cambiamos la contraseña en `etc/zabbix/zabbix_server.conf`. Descomentamos la línea pertinente y escribimos la contraseña elegida.

```
# DBPassword=password
```

- Configuramos `/etc/zabbix/apache.conf` y escogemos la timezone adecuada.

```
# php_value date.timezone Europe/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
    </Directory>

30,8 Top
```

- Finalmente, arrancamos los servicios con `systemctl`

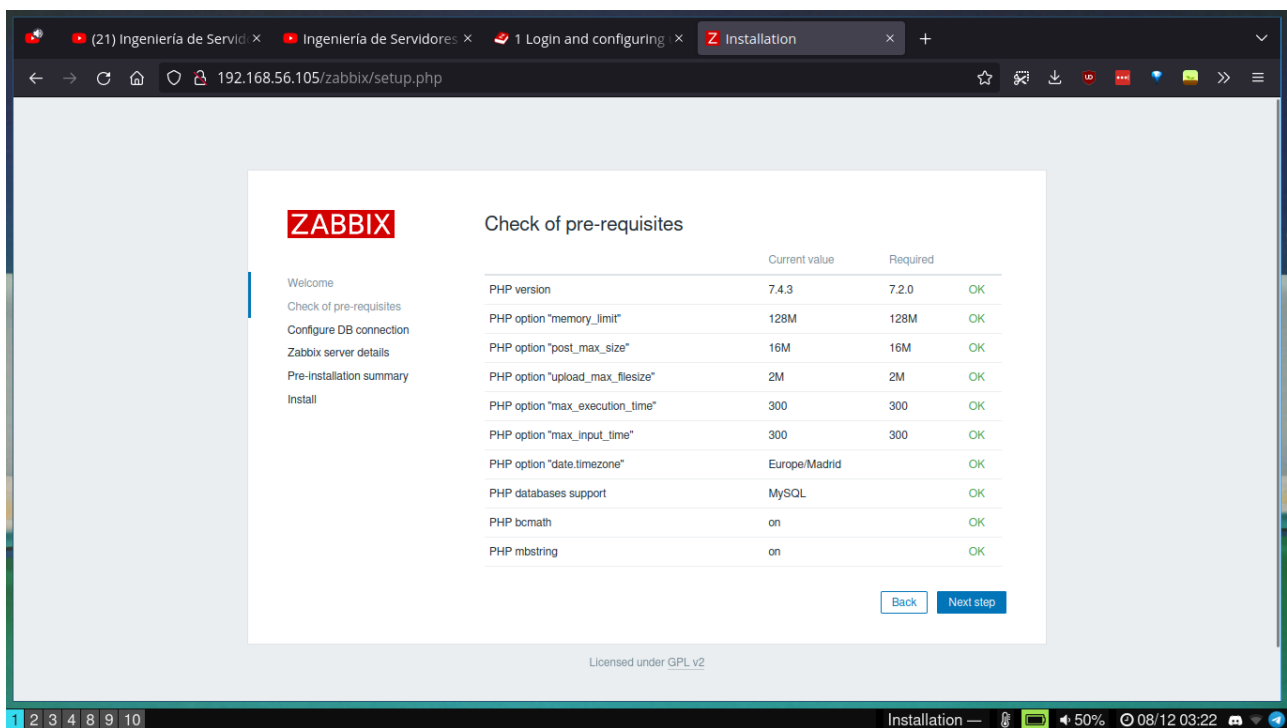
```
systemctl restart zabbix-server zabbix-agent apache2
systemctl enable zabbix-server zabbix-agent apache2
```

```
Query OK, 0 rows affected (0.00 sec)

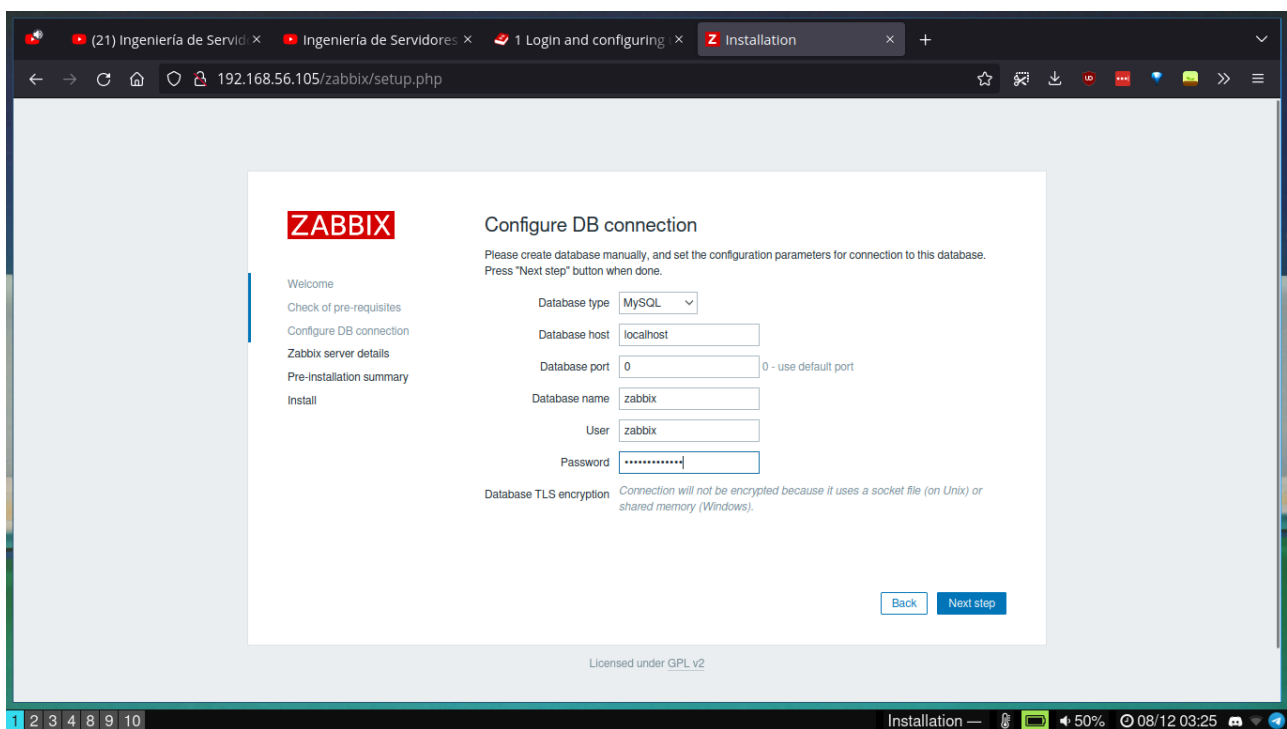
mysql> quit;
Bye
clararl@practicase:~$ sudo vi /etc/zabbix/zabbix_server.conf
clararl@practicase:~$ sudo vi /etc/z
zabbix/
clararl@practicase:~$ sudo vi /etc/zabbix/zabbix_server.conf
clararl@practicase:~$ ^Cdo vi /etc/zabbix/zabbix_server.conf
clararl@practicase:~$ sudo vi /etc/zabbix/apache.conf
clararl@practicase:~$ ^C:
clararl@practicase:~$ systemctl restart zabbix-server zabbix-agent apache2
== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ==
Authentication is required to restart 'zabbix-server.service'.
Authenticating as: clararl
Password:
== AUTHENTICATION COMPLETE ==
== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ==
Authentication is required to restart 'zabbix-agent.service'.
Authenticating as: clararl
Password:
== AUTHENTICATION COMPLETE ==
== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ==
Authentication is required to restart 'apache2.service'.
Authenticating as: clararl
Password:
== AUTHENTICATION COMPLETE ==
clararl@practicase:~$ 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-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/system/zabbix-server.service.
clararl@practicase:~$
```

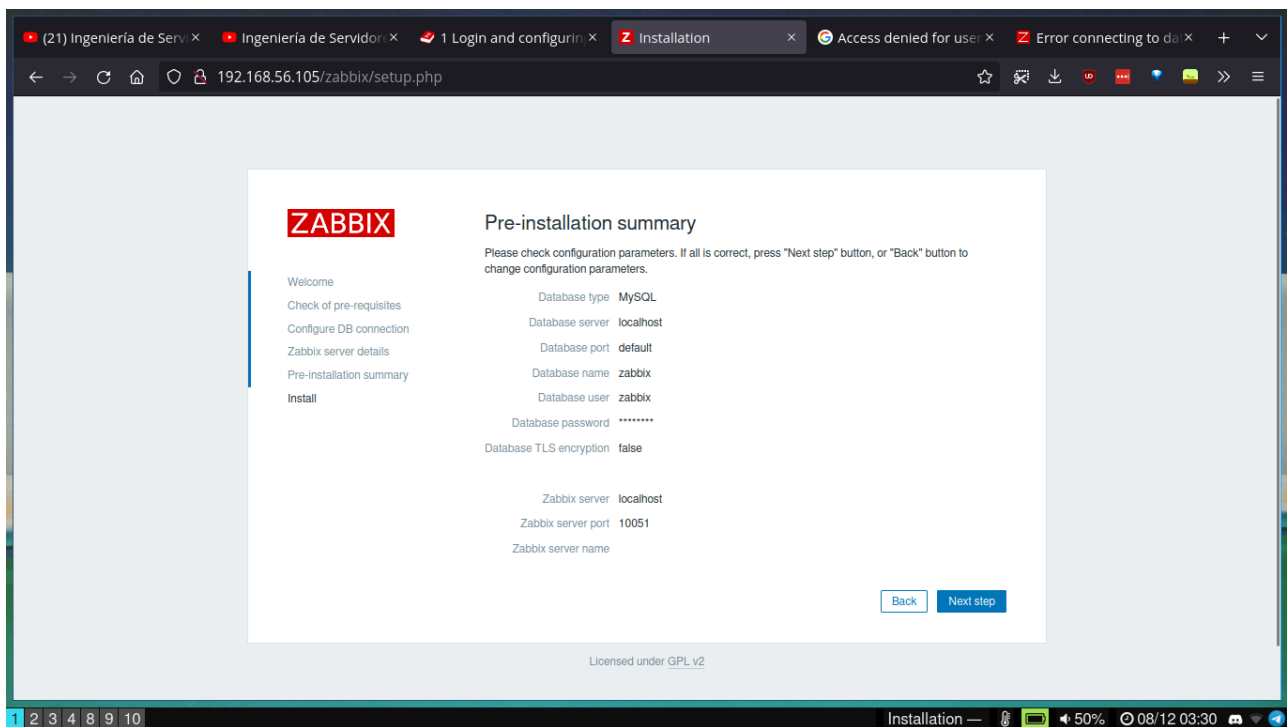
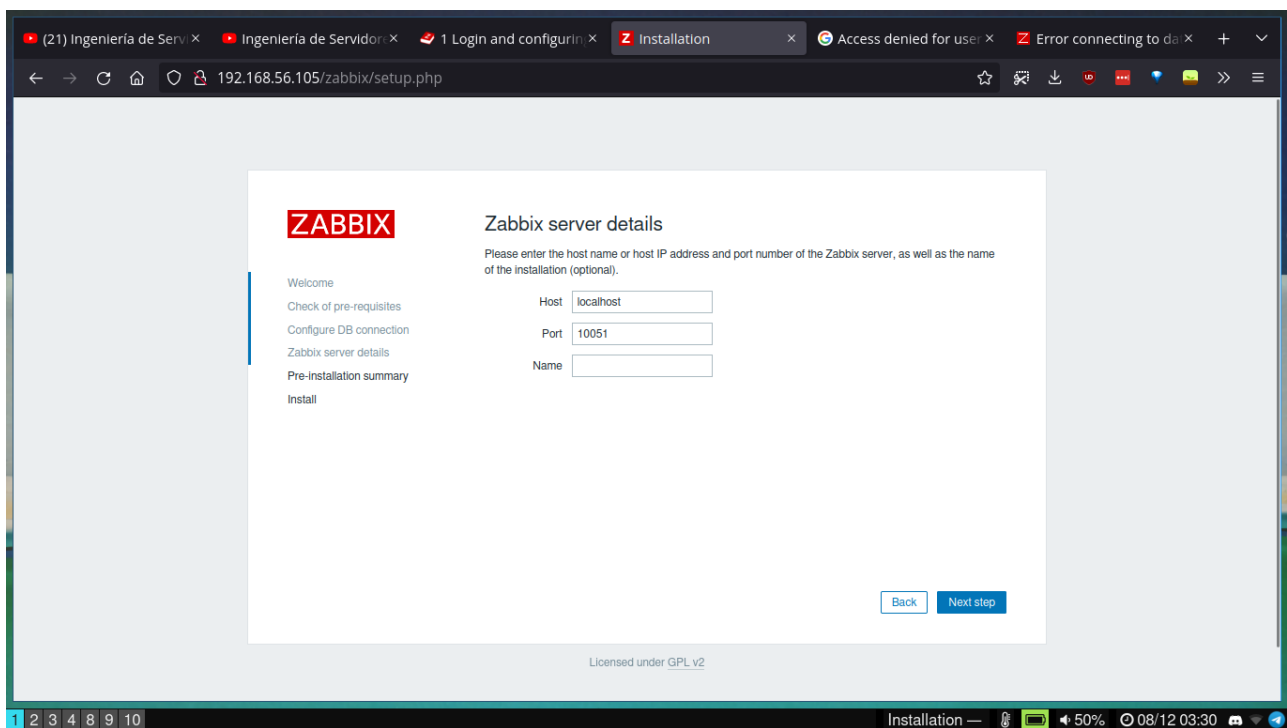
Inicio rápido

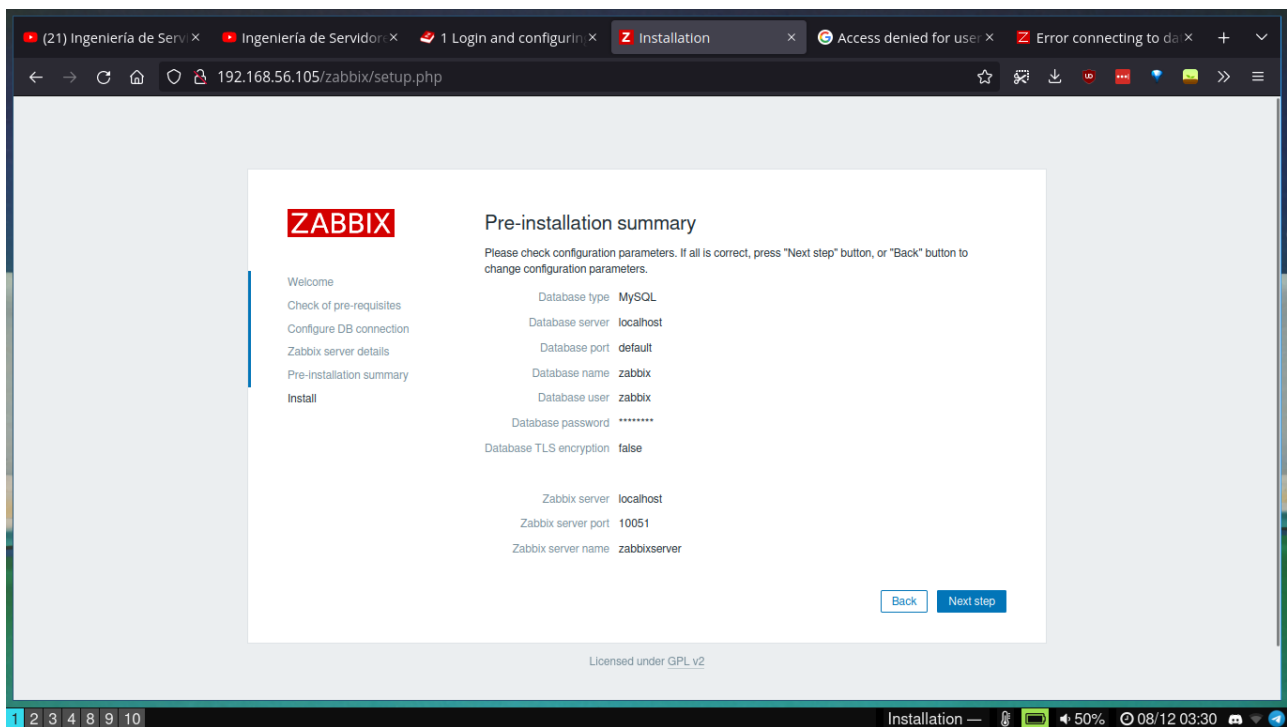
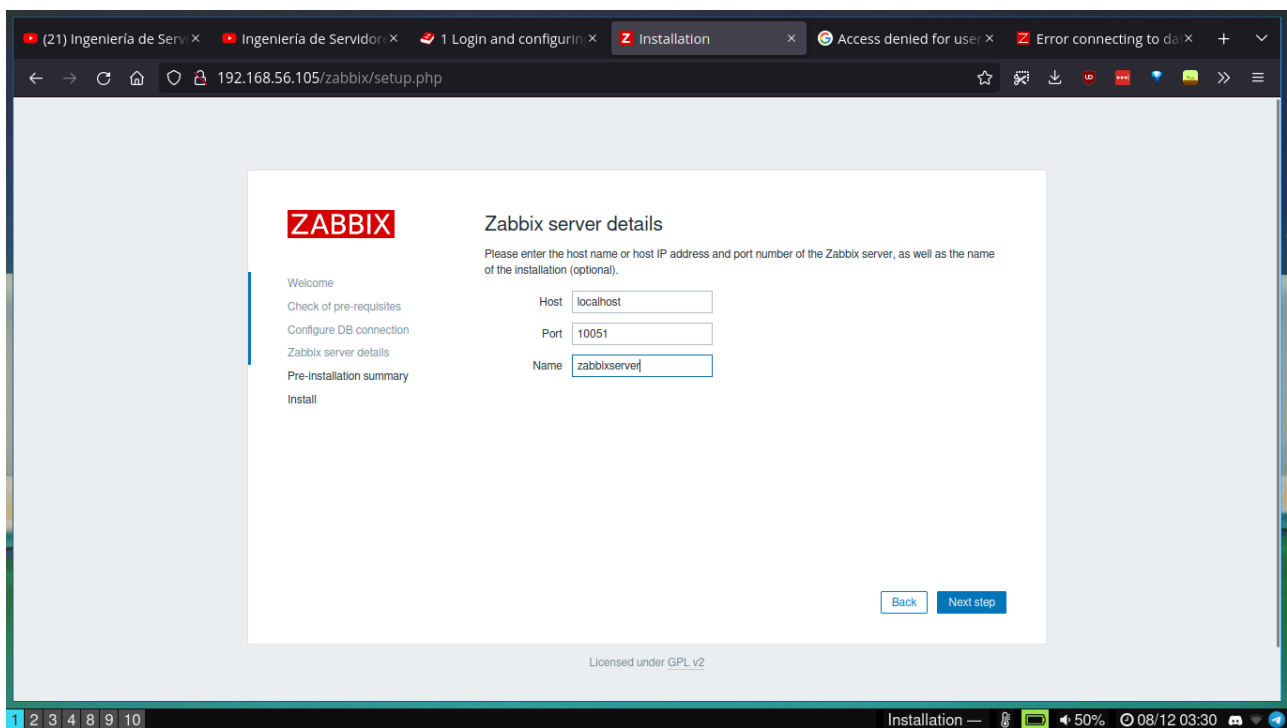
Primero [accedemos a la interfaz web de Zabbix](#) mediante un navegador, accediendo a 192.168.56.105/zabbix (la IP correspondiente a UbuntuServer)

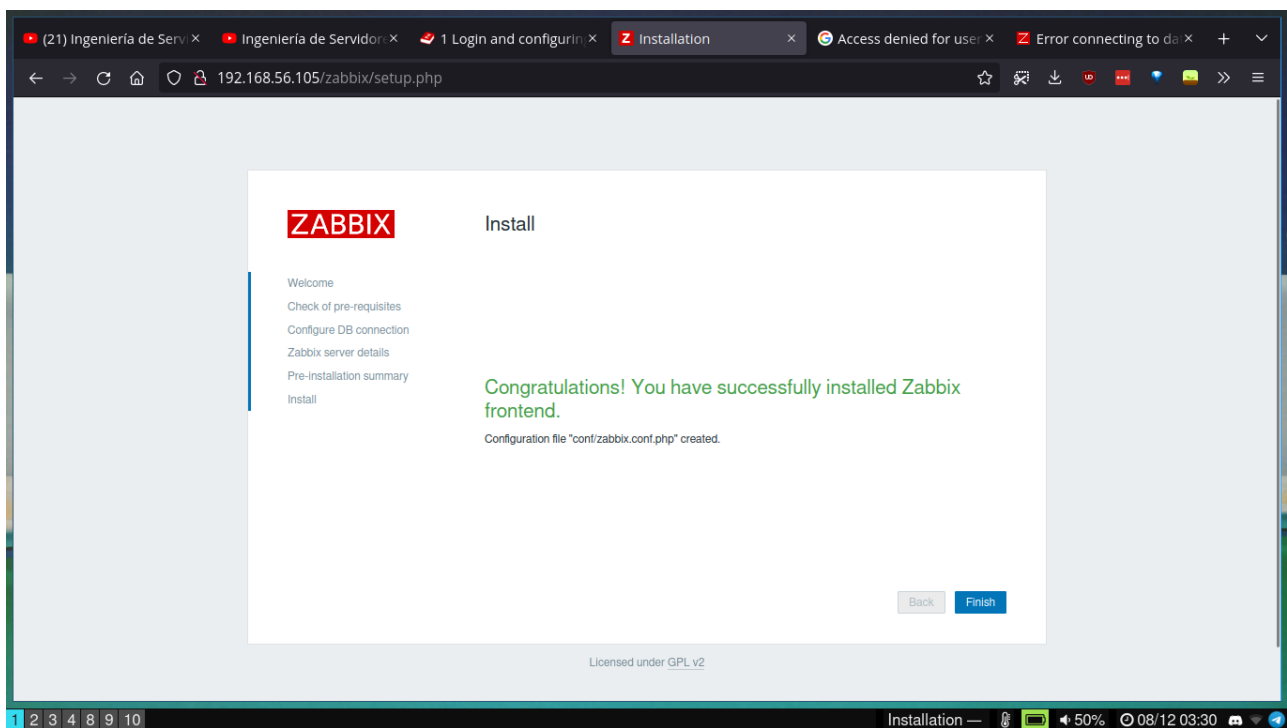


Y seguimos la [guía de inicio rápido](#) provista en la documentación:



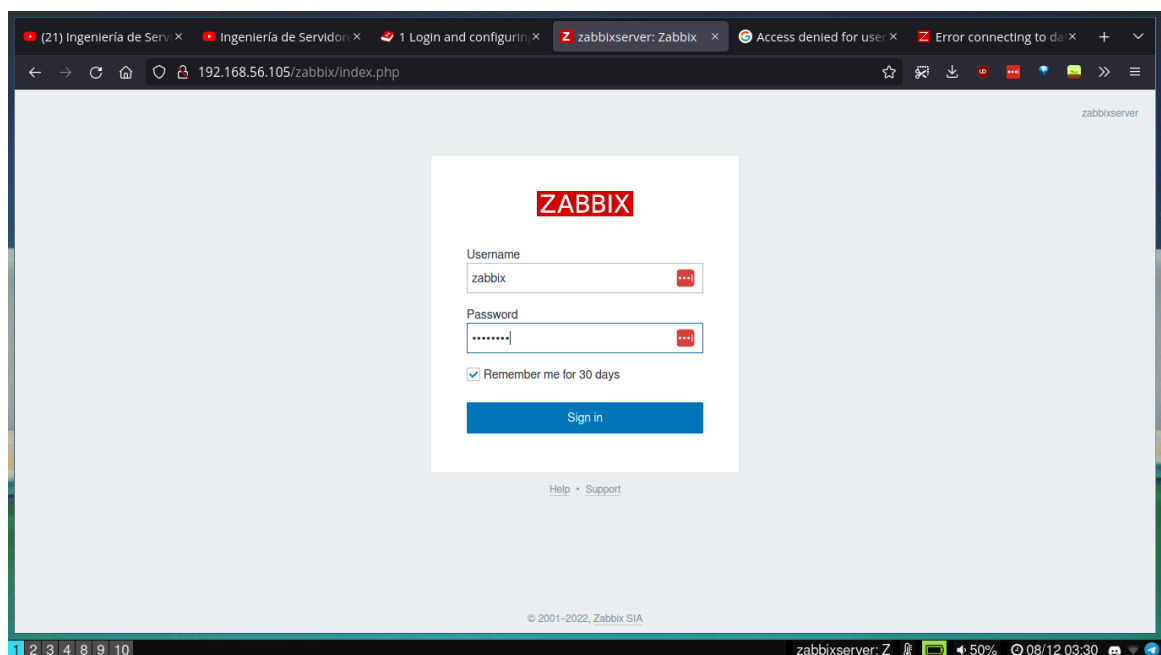






Continuamos con la configuración de nuestro ejercicio:

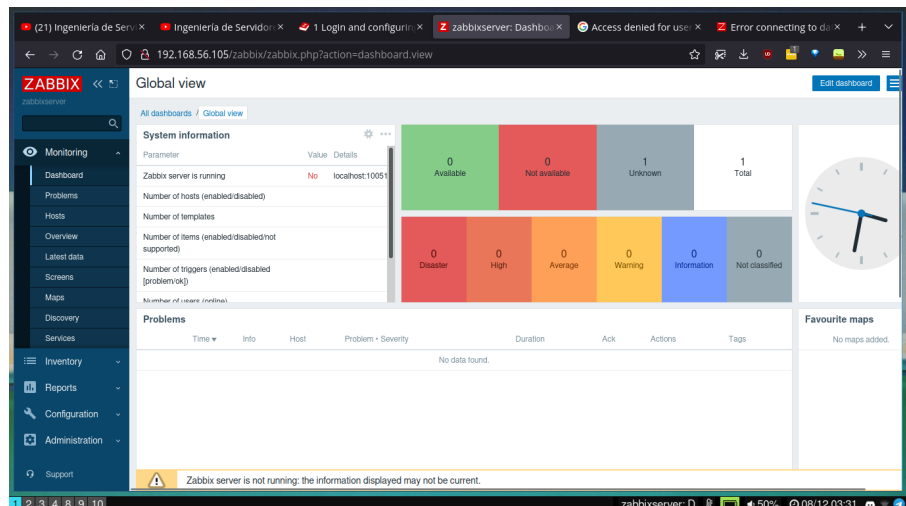
- Introducimos las credenciales, incluyendo la contraseña tal y como la configuramos anteriormente:



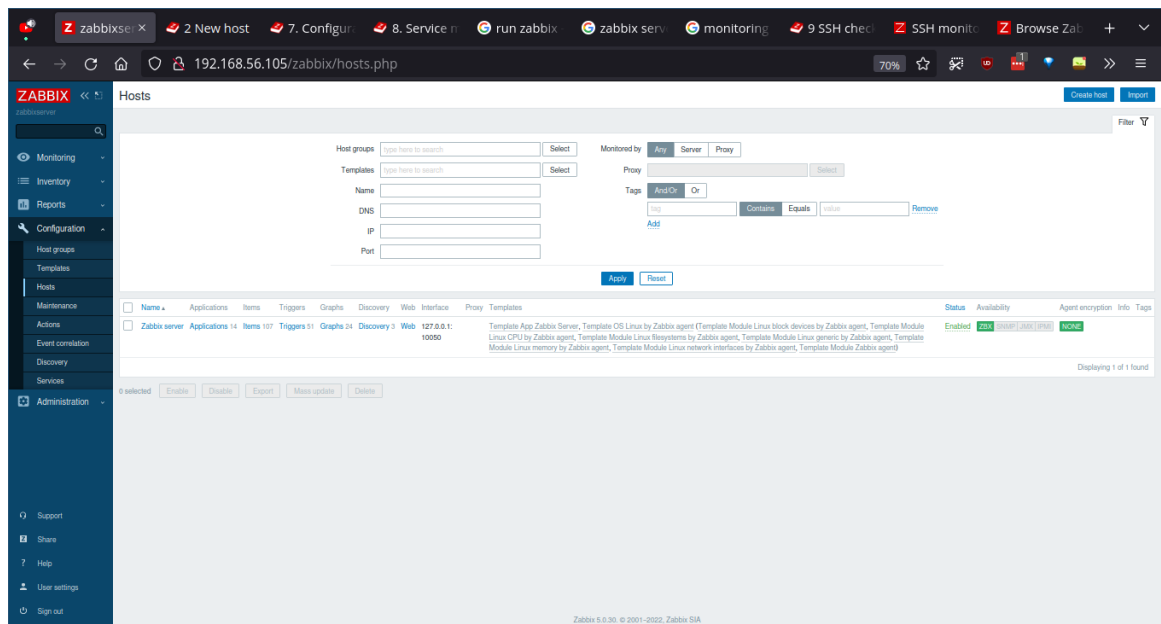
- Este paso tuvo un par de incidencias:
 - El servicio de zabbiz no iniciaba (En el panel de control: `Zabbix server is running: No`). Se solucionó profundizando en el concepto del server e iniciándolo:

```
service zabbix-server start
```

Además, en el archivo de configuración hubo una errata con la contraseña, por lo que los primeros login no tuvieron éxito. Finalmente, pudimos acceder al panel de control.



- Para monitorizarnos a nosotros mismos, tenemos que crear un nuevo host. Nos vamos a configuración, hosts y create host.



- Todos los cambios hasta el momento se han de reflejar en el archivo de configuración del agente de Zabbix `/etc/zabbix/zabbix_agentd.conf`.
 - Añadimos la ip del server desde el que vamos a *trackear* (la de ubuntu, puesto que se está monitorizando a sí mismo). Está en la parte del archivo correspondiente a server.
 - En la parte de host name, ponemos el nombre que hemos puesto en la interfaz web (UbuntuServer).
 - Aplicamos los cambios reiniciando los servicios.

Instalación rocky

De nuevo, seguiremos las instrucciones que nos proporciona la [página de Zabbix según nuestros parámetros](#).

- Instalar el repositorio de Zabbix, pero primero deberíamos deshabilitar los paquetes de Zabbix que nos ofrezca yum por precaución.

```
excludepkgs=zabbix*
```

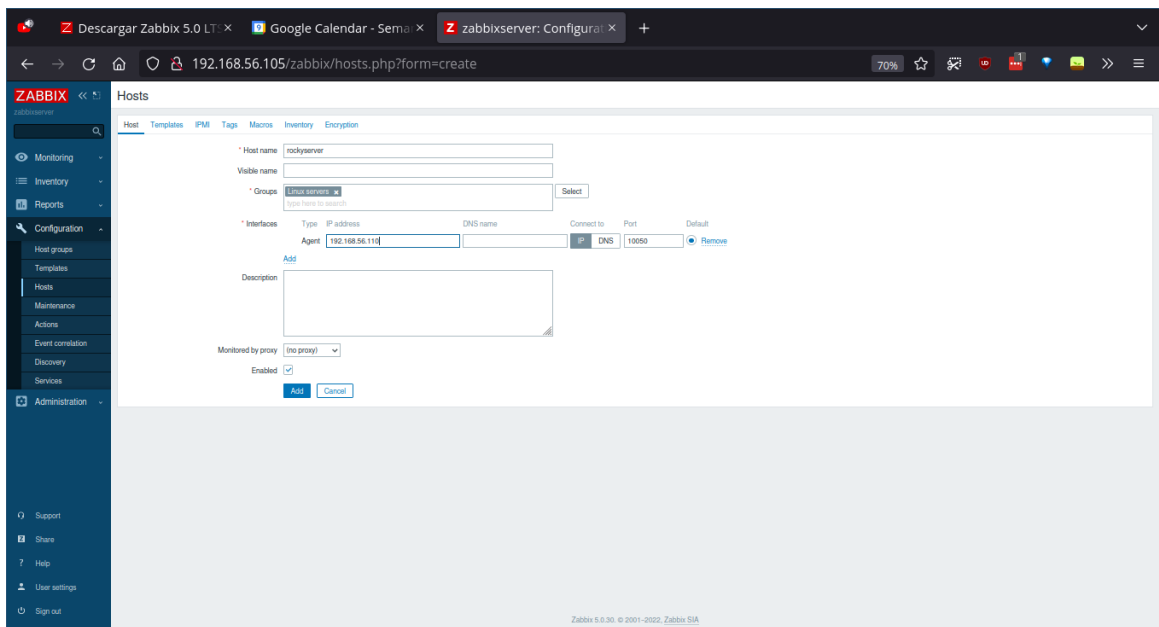
```
rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
```

```
dnf clean all
```

```
~
> ssh 192.168.56.110 -l clararl -p 22022
ssh: connect to host 192.168.56.110 port 22022: No route to host

~
> ssh 192.168.56.110 -l clararl
The authenticity of host '192.168.56.110 (192.168.56.110)' can't be established.
ED25519 key fingerprint is SHA256:uhAhfFcyGspS/QVWo/yF8kRNED+4DRcHH/a4QmJCUXg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.56.110' (ED25519) to the list of known hosts.
clararl@192.168.56.110's password:
Last login: Fri Dec 9 03:20:09 2022
[clararl@localhost ~]$ ls
holarocky
[clararl@localhost ~]$ vi /etc/yum.repos.d/epel.repo
[clararl@localhost ~]$ sudo !!
sudo vi /etc/yum.repos.d/epel.repo
[sudo] password for clararl:
Sorry, try again.
[sudo] password for clararl:
[clararl@localhost ~]$ rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
warning: /var/tmp/rpm-tmp.f81GGq: Header V4 RSA/SHA512 Signature, key ID 08efa7dd: NOKEY
error: can't create transaction lock on /var/lib/rpm/.rpm.lock (Permission denied)
[clararl@localhost ~]$ sudo !!
sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/5.0/rhel/9/x86_64/zabbix-release-5.0-3.el9.noarch.rpm
warning: /var/tmp/rpm-tmp.nUXX0m: Header V4 RSA/SHA512 Signature, key ID 08efa7dd: NOKEY
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:zabbix-release-5.0-3.el9 ##### [100%]
[clararl@localhost ~]$
```

- Creamos un host mediante la interfaz web, en el grupo linux servers y con la IP de Rocky. Lo llamamos RockyServer.

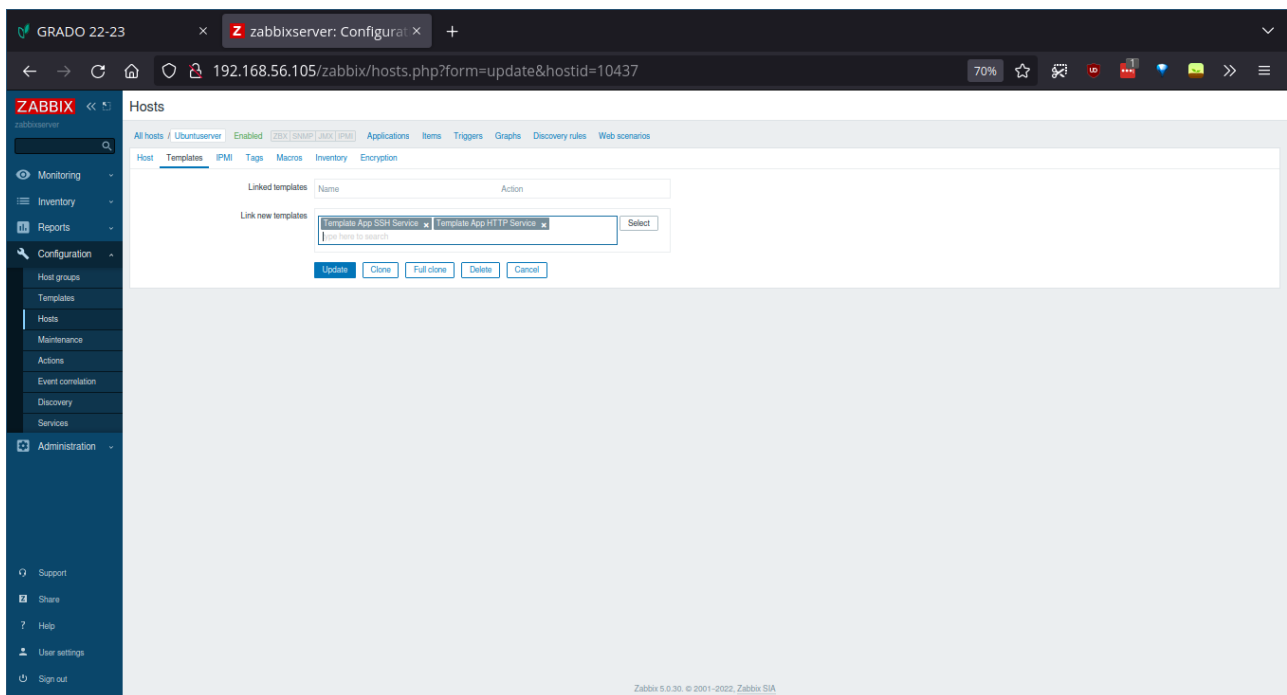
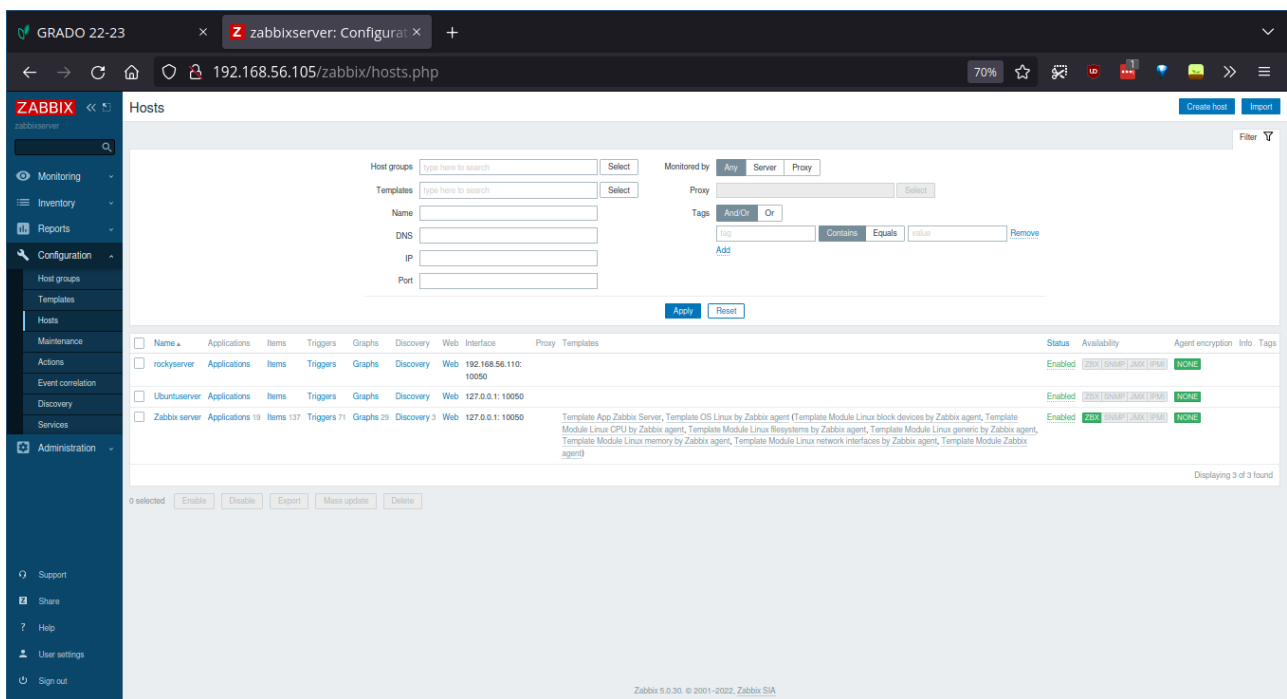


- En el archivo de configuración del agente de Zabbix `/etc/zabbix/zabbix_agentd.conf` añadimos la IP de Rocky. En la parte de *host name*, ponemos el nombre que hemos puesto en la interfaz web (RockyServer).
- Aplicamos los cambios reiniciando los servicios
- Finalmente, configuramos el firewall para habilitar los puertos que vamos a emplear

```
> sudo firewall-cmd --add-port=10050/tcp --permanent
> sudo firewall-cmd --add-port=22/tcp --permanent
> sudo firewall-cmd --add-port=80/tcp --permanent
> sudo firewall-cmd --reload
```

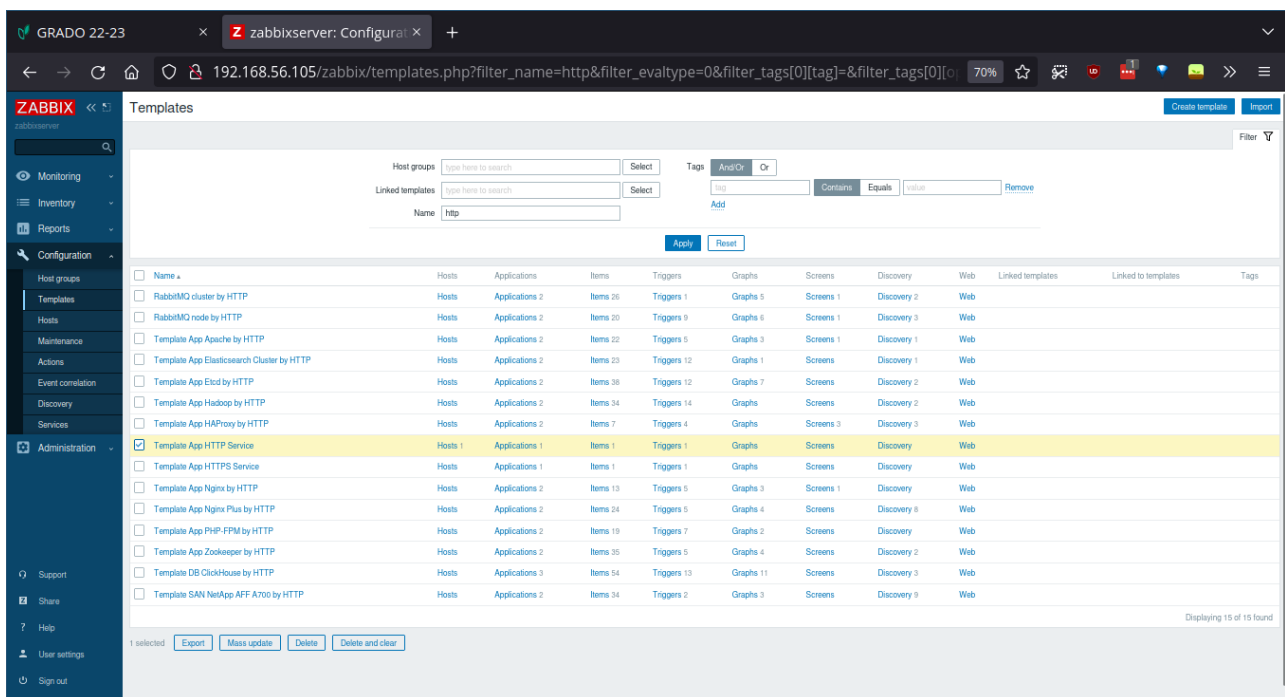
Monitorización de SSH y HTTP

Nos vamos a Configuración>Hosts y entramos al host UbuntuServer. Seleccionamos la opción Templates, y añadimos las de ssh y http.



Accedemos al menú Configuración>Templates, y buscamos las Templates recién añadidas. Seleccionamos Items, y añadimos las siguientes keys:

- En SSH: `net.tcp.service[ssh,,22022]`
- En HTTP: `net.tcp.service[http,,80]`



Para comprobar si realmente está funcionando la monitorización, instalaremos y ejecutaremos `zabbix-get`.

Durante la instalación de `zabbix-get`, hemos tenido **problemas con el DNS**. Estos se resolvieron añadiendo el server DNS de Google al archivo `/etc/resolv.conf` y reiniciando el servicio `systemd-resolved`.

Ejecutamos `zabbix-get` pasando como parámetros la IP, el puerto (10050, el que asignamos a Zabbix) y el servicio. Un 0 indica que el servicio está caído, y un 1 que está activo.

```
> zabbix_get -s 192.168.56.105 -p 10050 -k net.tcp.service[ssh,,22022]
> zabbix_get -s 192.168.56.105 -p 10050 -k net.tcp.service[http,,80]
> zabbix_get -s 192.168.56.110 -p 10050 -k net.tcp.service[ssh]
> zabbix_get -s 192.168.56.110 -p 10050 -k net.tcp.service[http]
```

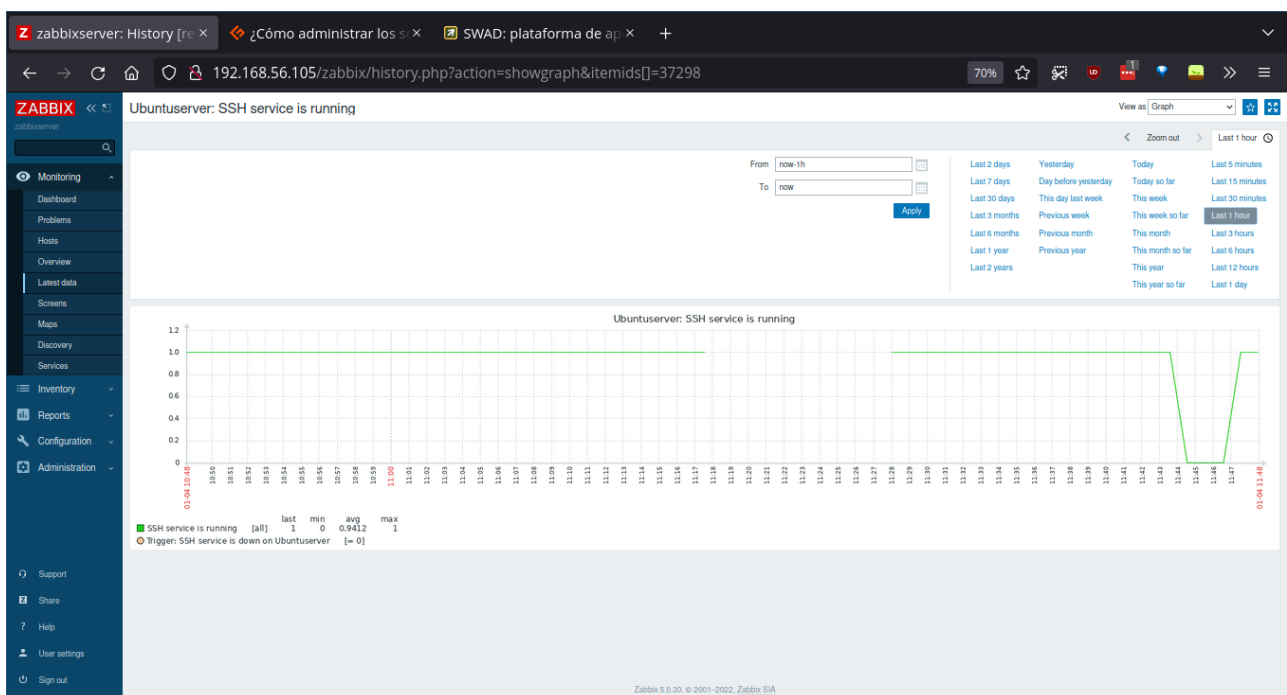
```

clararl@practicasis:~$ systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2023-01-04 09:23:32 UTC; 1h 19min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 141976 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
    Main PID: 141987 (sshd)
      Tasks: 1 (limit: 1066)
     Memory: 5.6M
    CGroup: /system.slice/ssh.service
            └─141987 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jan 04 10:34:38 practicasis sshd[145235]: Connection closed by 127.0.0.1 port 48308 [preauth]
Jan 04 10:35:39 practicasis sshd[145253]: Connection closed by 127.0.0.1 port 34720 [preauth]
Jan 04 10:36:38 practicasis sshd[145270]: Connection closed by 127.0.0.1 port 45566 [preauth]
Jan 04 10:37:38 practicasis sshd[145287]: Connection closed by 127.0.0.1 port 60150 [preauth]
Jan 04 10:38:38 practicasis sshd[145304]: Connection closed by 127.0.0.1 port 37672 [preauth]
Jan 04 10:39:38 practicasis sshd[145335]: Connection closed by 127.0.0.1 port 34738 [preauth]
Jan 04 10:40:38 practicasis sshd[145352]: Connection closed by 127.0.0.1 port 48936 [preauth]
Jan 04 10:41:38 practicasis sshd[145370]: Connection closed by 127.0.0.1 port 41506 [preauth]
Jan 04 10:42:38 practicasis sshd[145391]: Connection closed by 127.0.0.1 port 39052 [preauth]
Jan 04 10:42:51 practicasis sshd[145394]: Connection closed by 127.0.0.1 port 50152 [preauth]
clararl@practicasis:~$ zabbix_get -s 192.168.56.105 -p 10050 -k net.tcp.service[ssh,,22022]
1
clararl@practicasis:~$ zabbix_get -s 192.168.56.105 -p 10050 -k net.tcp.service[http,,80]
1
clararl@practicasis:~$ zabbix_get -s 192.168.56.110 -p 10050 -k net.tcp.service[ssh]
1
clararl@practicasis:~$ zabbix_get -s 192.168.56.110 -p 10050 -k net.tcp.service[http]
1
clararl@practicasis:~$ systemctl stop ssh
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to stop 'ssh.service'.
Authenticating as: clararl
Password:

```

Podemos tirar y reiniciar un servicio y comprobar desde el panel de control de Zabbix la gráfica de monitorización del servicio, en este caso, SSH desde UbuntuServer:



Ejercicio 2: Ansible

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 como, por ejemplo,

asegurarse de que tenemos la última versión instalada de `httpd` y que está en ejecución.

Instalación de Ansible y conexión con las máquinas virtuales

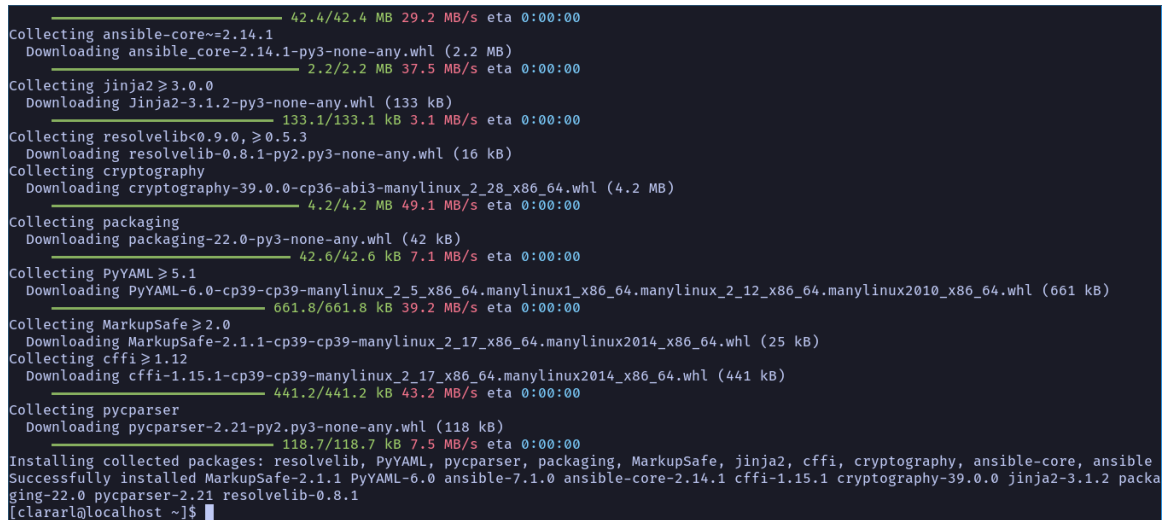
Seguiremos la [guía de instalación](#) provista en la documentación oficial.

- Primero tenemos que comprobar que Python está instalado tanto en la máquina host (Ubuntu Server) como en los nodos que vamos a configurar. Podemos comprobarlo accediendo a la consola de python ejecutando `python` o `python3`.
- Además, necesitaremos el módulo `pip`. Para instalarlo:

```
> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
> python3 get-pip.py --user
```

- A continuación, instalaremos el paquete Ansible

```
> sudo apt install ansible
```



```
Collecting ansible-core==2.14.1
  Downloading ansible_core-2.14.1-py3-none-any.whl (2.2 MB)
    42.4/42.4 MB 29.2 MB/s eta 0:00:00
Collecting jinja2>=3.0.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    2.2/2.2 MB 37.5 MB/s eta 0:00:00
Collecting resolvelib<0.9.0, >=0.5.3
  Downloading resolvelib-0.8.1-py2.py3-none-any.whl (16 kB)
    133.1/133.1 kB 3.1 MB/s eta 0:00:00
Collecting cryptography
  Downloading cryptography-39.0.0-cp36-abi3-manylinux_2_28_x86_64.whl (4.2 MB)
    4.2/4.2 MB 49.1 MB/s eta 0:00:00
Collecting packaging
  Downloading packaging-22.0-py3-none-any.whl (42 kB)
    42.6/42.6 kB 7.1 MB/s eta 0:00:00
Collecting PyYAML >=5.1
  Downloading PyYAML-6.0-cp39-cp39-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux2010_x86_64.whl (661 kB)
    661.8/661.8 kB 39.2 MB/s eta 0:00:00
Collecting MarkupSafe >=2.0
  Downloading MarkupSafe-2.1.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
    441.2/441.2 kB 43.2 MB/s eta 0:00:00
Collecting cffi >=1.12
  Downloading cffi-1.15.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (441 kB)
    441.2/441.2 kB 43.2 MB/s eta 0:00:00
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
    118.7/118.7 kB 7.5 MB/s eta 0:00:00
Installing collected packages: resolvelib, PyYAML, pycparser, packaging, MarkupSafe, jinja2, cffi, cryptography, ansible-core, ansible
Successfully installed MarkupSafe-2.1.1 PyYAML-6.0 ansible-7.1.0 ansible-core-2.14.1 cffi-1.15.1 cryptography-39.0.0 jinja2-3.1.2 packa
ging-22.0 pycparser-2.21 resolvelib-0.8.1
[clarar@localhost ~]$
```

- También instalamos el módulo `python` de Ansible mediante `pip`

```
> python3 -m pip install --user ansible
```



```
Installing collected packages: wheel, pip
WARNING: The script wheel is installed in '/home/clararl/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts pip, pip3 and pip3.8 are installed in '/home/clararl/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-22.3.1 wheel-0.38.4
clararl@practicasis:~$ python3 get-pip.py --user --no-warn-script-location.

Usage:
  get-pip.py install [options] <requirement specifier> [package-index-options] ...
  get-pip.py install [options] -r <requirements file> [package-index-options] ...
  get-pip.py install [options] [-e] <vcs project url> ...
  get-pip.py install [options] [-e] <local project path> ...
  get-pip.py install [options] <archive url/path> ...

no such option: --no-warn-script-location.
clararl@practicasis:~$ --no-warn-script-location
--no-warn-script-location: command not found
clararl@practicasis:~$ python3 get-pip.py --user --no-warn-script-location
Collecting pip
  Using cached pip-22.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting to uninstall: pip
    Found existing installation: pip 22.3.1
    Uninstalling pip-22.3.1:
      Successfully uninstalled pip-22.3.1
  Successfully installed pip-22.3.1
clararl@practicasis:~$

~
> ssh 192.168.56.110 -l clararl -p 22022
ssh: connect to host 192.168.56.110 port 22022: No route to host

~
> ssh 192.168.56.110 -l clararl
clararl@192.168.56.110's password:
Last login: Fri Dec 9 04:17:18 2022 from 192.168.56.1
[clararl@localhost ~]$ ls
holarocky
[clararl@localhost ~]$ python3 -m pip -v
/usr/bin/python3: No module named pip
[clararl@localhost ~]$ curl https://bootstrap.pypa.io/get-pip.py -o-
get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  T
ime  Current              Dload  Upload    Total   Spent    L
eft  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:
100 2509k 100 2509k    0     0 8622k      0 --:--:-- --:--:-- --:
--:-- 8622k
[clararl@localhost ~]$ python3 get-pip.py --user
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
    2.1/2.1 MB 18.4 MB/s eta 0:00:00
Collecting setuptools
  Downloading setuptools-65.6.3-py3-none-any.whl (1.2 MB)
    1.2/1.2 MB 40.4 MB/s eta 0:00:00
Collecting wheel
  Downloading wheel-0.38.4-py3-none-any.whl (36 kB)
Installing collected packages: wheel, setuptools, pip
Successfully installed pip-22.3.1 setuptools-65.6.3 wheel-0.38.4
[clararl@localhost ~]$
```

- Durante la instalación de pip y Ansible hemos tenido un warning que indica que la ruta en la que se instalan los módulos no pertenece al path de Python. Podemos ignorar el warning y continuar con la tarea, ya que no es un problema grave: simplemente, tendremos que indicar a Python la ruta completa para usar los módulos.

Una vez instalado, podemos continuar con la [guía de inicio](#) de Ansible. Primero nos familiarizamos con los términos básicos de Ansible, que son el **nodo de control** (la máquina que tiene Ansible), **nodo controlado** (un sistema remoto o el propio host que es controlado por Ansible) e **inventario** (una lista de nodos controlados).

- Creamos un inventario. Lo hacemos añadiendo en el nodo de control las IP de los nodos controlados, Ubuntu Server y Rocky, en el archivo `/etc/ansible/hosts`

```
[miinventario]
192.168.56.105
192.168.56.110
```

- Podemos comprobar que se han añadido correctamente listando los host:

```
> sudo ansible all --list-hosts
```

```
clararl@practicasis:~$ sudo ansible all --list-hosts
hosts (2):
  192.168.56.105
  192.168.56.110
clararl@practicasis:~$
```

- A continuación, vamos a configurar la conexión SSH entre el nodo de control y los nodos controlados. Desde el nodo controlador, generamos y añadimos las claves públicas de cada nodo a las `authorized_keys`:

- Nos topamos con que nos falta espacio en el home. Vamos a extender el /home con un disco virtual de 15GB (el cual hemos creado desde VirtualBox)

- Comprobamos primero el estado actual de nuestras particiones:

```

clarar@practicasiser:~$ lsblk -f

```

| NAME | FSTYPE | LABEL | UUID | FSAVAIL | FSUSEX | MOUNTPOINT |
|-----------------------|-------------------|-----------------|---------------------------------------|---------|--------|-------------------|
| loop0 | squashfs | | | 0 | 100% | /snap/snapd/17883 |
| loop1 | squashfs | | | 0 | 100% | /snap/snapd/17950 |
| loop2 | squashfs | | | 0 | 100% | /snap/1xd/16099 |
| loop3 | squashfs | | | 0 | 100% | /snap/1xd/24061 |
| loop4 | squashfs | | | 0 | 100% | /snap/core20/171 |
| loop5 | squashfs | | | 0 | 100% | /snap/core20/1738 |
| loop6 | squashfs | | | 0 | 100% | /snap/core18/2632 |
| loop7 | squashfs | | | 0 | 100% | /snap/core18/2667 |
| sda | | | | | | |
| ├ sda1 | | | | | | |
| ├ sda2 | linux_raid_member | ubuntu-server:0 | eF850708-73e2-d649-9adb-dc2876be00a0 | | | |
| └ md0 | ext4 | | b39cfedf-6c30-4 | | | |
| a05-872b-7de4dac60e34 | 40.1M | 82% /boot | | | | |
| ├ md1 | linux_raid_member | ubuntu-server:1 | b8945edf-6e62-5010-2771-2636fdd0d0fa | | | |
| ├ md1_crypt-0 | crypto_LUKS | | eefc4b5-781b-41a0-8295-5531ae854bc9 | | | |
| ├ LVM2_member | | | QcFLW-470t-hJNS-LZ17-hCcx-7f6n-13ZPVE | | | |
| ├ vgo-swap | swap | | 50e1e6ef-d77e-47ca-b9c6-e93c1c608b9a | | | [SWAP] |
| ├ vgo-hogar | ext4 | | a744192b-a95f-48e4-9a59-2829f74f02aa | 0 | 93% | /home |
| └ vgo-raid | ext4 | | d81564de-09b0-40cb-bbb6-93c3e27ad069 | 1.6G | 74% | / |
| sdb | | | | | | |
| ├ sdb1 | | | | | | |
| ├ sdb2 | linux_raid_member | ubuntu-server:0 | eF850708-73e2-d649-9adb-dc2876be00a0 | | | |
| └ md0 | ext4 | | b39cfedf-6c30-4a05-872b-7de4dac60e34 | 40.1M | 82% | /boot |
| ├ sdb3 | linux_raid_member | ubuntu-server:1 | b8945edf-6e62-5010-2771-2636fdd0d0fa | | | |
| ├ md1 | crypto_LUKS | | eefc4b5-781b-41a0-8295-5531ae854bc9 | | | |
| ├ md1_crypt-0 | LVM2_member | | QcFLW-470t-hJNS-LZ17-hCcx-7f6n-13ZPVE | | | |
| ├ vgo-swap | swap | | 50e1e6ef-d77e-47ca-b9c6-e93c1c608b9a | | | |
| ├ vgo-hogar | ext4 | | a744192b-a95f-48e4-9a59-2829f74f02aa | 0 | 93% | /home |
| └ vgo-raid | ext4 | | d81564de-09b0-40cb-bbb6-93c3e27ad069 | 1.6G | 74% | / |
| sdc | | | | | | |

```

clarar@practicasiser:~$ sudo vgdisplay
[sudo] password for clarar:
  --- Volume group ---
 VG Name       vgo

```

- Extendemos el grupo de volúmenes vg0, y luego el volumen lógico que queremos ampliar, `/dev/vg0/hogar`

```
> sudo vgextend vg0 /dev/sdc
> sudo lvextend -l +100%FREE /dev/vg0/hogar
```

```
clararl@practicasis:~$ sudo vgextend vg0 /dev/sdc
Physical volume "/dev/sdc" successfully created.
Volume group "vg0" successfully extended
```

```
clararl@practicasis:~$ sudo lvextend -l +100%FREE /dev/vg0/hogar
Size of logical volume vg0/hogar changed from 500.00 MiB (125 extents) to 15.48 GiB (3964 extents).
Logical volume vg0/hogar successfully resized.
```

- Finalmente hacemos el resize

```
> sudo resize2fs /dev/vg0/hogar
```

Y ya habríamos expandido nuestro LVM. Podemos continuar con las claves SSH.

```
Filesystem at /dev/vg0/hogar is mounted on /home; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 2
The filesystem on /dev/vg0/hogar is now 4059136 (4k) blocks long.

clararl@practicasis:~$ lsblk -f
```

| NAME | FSTYPE | LABEL | UUID | FSAVAIL | FSUSE% | MOUNTPOINT |
|---------------------------|-------------------|-----------------|--|---------|--------|-------------------|
| loop0 | squashfs | | | 0 | 100% | /snap/snapd/17883 |
| loop1 | squashfs | | | 0 | 100% | /snap/snapd/17950 |
| loop2 | squashfs | | | 0 | 100% | /snap/lxd/16099 |
| loop3 | squashfs | | | 0 | 100% | /snap/lxd/24061 |
| loop4 | squashfs | | | 0 | 100% | /snap/core20/1778 |
| loop5 | squashfs | | | 0 | 100% | /snap/core20/1738 |
| loop6 | squashfs | | | 0 | 100% | /snap/core18/2632 |
| loop7 | squashfs | | | 0 | 100% | /snap/core18/2667 |
| sda | | | | | | |
| ├─sda1 | | | | | | |
| ├─sda2 | linux_raid_member | ubuntu-server:0 | ef850708-73e2-d649-9adb-dc2876be00a0 | | | |
| ├─└─lmd0 | ext4 | | b39c6fed-6c30-4a05-872b-7de4dac66e34 | 40.1M | 82% | /boot |
| ├─sda3 | linux_raid_member | ubuntu-server:1 | b8945edf-6e62-5 | | | |
| ├─└─010-2771-263f6fd0d0fa | | | | | | |
| ├─└─└─lmd1 | crypto_LUKS | | eefc4cb5-781b-41a0-8295-5531ae854bc9 | | | |
| ├─└─└─└─dm_crypt-0 | LVM2_member | | QCfLWB-470t-hJNS-LZI7-hCcx-7F6n-13ZPVE | | | |
| ├─└─└─└─└─vg0-swap | swap | | 50e1e66f-d77e-47ca-b9c6-a93c1c68b69a | | | [SWAP] |
| ├─└─└─└─└─└─vg0-hogar | ext4 | | a744192b-a95f-48ea-9a59-2829f4f02aea | 14G | 3% | /home |
| ├─└─└─└─└─└─└─vg0-raiz | ext4 | | d81564de-09b0-40cb-bbb6-93c3e27ad069 | 1.6G | 74% | / |
| sdb | | | | | | |
| ├─sdb1 | | | | | | |
| ├─sdb2 | | | | | | |
| ├─└─lmd0 | linux_raid_member | ubuntu-server:0 | ef850708-73e2-d649-9adb-dc2876be00a0 | | | |
| ├─└─└─lmd1 | ext4 | | b39c6fed-6c30-4a05-872b-7de4dac66e34 | 40.1M | 82% | /boot |
| ├─sdb3 | linux_raid_member | ubuntu-server:1 | b8945edf-6e62-5010-2771-263f6fd0d0fa | | | |
| ├─└─└─lmd1 | crypto_LUKS | | eefc4cb5-781b-41a0-8295-5531ae854bc9 | | | |
| ├─└─└─└─dm_crypt-0 | LVM2_member | | QCfLWB-470t-hJNS-LZI7-hCcx-7F6n-13ZPVE | | | |
| ├─└─└─└─└─vg0-swap | swap | | 50e1e66f-d77e-47ca-b9c6-a93c1c68b69a | | | [SWAP] |
| ├─└─└─└─└─└─vg0-hogar | ext4 | | a744192b-a95f-48ea-9a59-2829f4f02aea | 14G | 3% | /home |
| ├─└─└─└─└─└─└─vg0-raiz | ext4 | | d81564de-09b0-40cb-bbb6-93c3e27ad069 | 1.6G | 74% | / |
| sdc | LVM2_member | | G43R0M-LDz0-w5Z4-EsL7-AL97-c8A3-dp0gPc | | | |

```

Your public key has been saved in /home/clararl/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:27gIctshZgm4DJrjeaqxa5D9sUSs3bLX+Xmbu+BzLlo clararl@practicasise
The key's randomart image is:
+--[RSA 3072]--+
|
| .
| o
|oo.+ . .S .
|*oo.+o +o
|B..oBo..o...E.
|.=.*o=.o.. o=.o.
|*=o . o.. ...**+
+--[SHA256]--+
clararl@practicasise:~$ ssh-copy-id 192.168.56.105 -p 22022 -f
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/clararl/.ssh/id_rsa.pub"
clararl@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.

clararl@practicasise:~$ ssh-copy-id 192.168.56.110
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/clararl/.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 the new keys
clararl@192.168.56.110's password:

Number of key(s) added: 1

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

clararl@practicasise:~$ █

```

- Finalmente, hacemos ping a todas las máquinas:

```
> ansible all -m ping
```

- Hemos tenido que modificar el archivo `/etc/ansible/hosts` para especificar el puerto 22022

```

clararl@practicasise:~$ ansible all -m ping
Enter passphrase for key '/home/clararl/.ssh/id_rsa': 192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}

192.168.56.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}

```