ZABBIX

Antes de empezar, la fuente de la cual he sacado la información sobre la instalación y uso de Zabbix es esta:

▷ Cómo instalar Zabbix en Ubuntu 20.04 LTS ✔ [2020] Paso a paso (comoinstalar.me)

Instalación en el host

El host será la máquina virtual de Ubuntu Server con la configuración de la práctica 2 (ssh y http puestos en marcha).

El primer paso será instalar el repositorio oficial de Zabbix para Ubuntu, por lo que descargamos el paquete que configura dicho repositorio.

```
javierrp@javierrp:~$ wget -q https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/za
bbix-release_5.0-1+focal_all.deb
javierrp@javierrp:~$ _
```

Por si la orden no se ve completa (por tener la ventana minimizada para seguir los pasos):

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

A continuación, lo instalamos con la herramienta dpkg:

```
javierrp@javierrp:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
[sudo] password for javierrp:
Selecting previously unselected package zabbix-release.
(Reading database ... 72456 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) ...
javierrp@javierrp:~$
```

Y actualizamos la información de los repositorios para incluir los nuevos paquetes de Zabbix.

```
javierrp@javierrp:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
Get:3 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4930 B]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [107 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [670 kB]
Get:7 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1192 B]
Get:8 http://es.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [167 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [11
Get:10 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [695
Get:11 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [3217 B]
Get:12 http://es.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [135
Get:13 http://es.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadat
Get:14 http://es.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [5
Get:15 http://es.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metad
Fetched 2023 kB in 3s (739 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
73 packages can be upgraded. Run 'apt list --upgradable' to see them.
javierrp@javierrp:~$ _
```

Ahora vamos a instalar Zabbix Server seleccionando los paquetes para el motor de bases de datos que nos interese. Como tenemos un entorno LAMP con Apache, PHP y MySQL, la elección adecuada es la siguiente:

```
javierrp@javierrp:~$ sudo apt install -y zabbix-server-mysql zabbix-frontend-php zabb
zabbix-agent
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core fping libfontconfig1 libgd3 libjbig0 libjpeg-tu
  libmysqlclient21 libodbc1 libonig5 libopenipmi0 libsensors-config libsensors5 libsr
  libsnmp35 libtiff5 libwebp6 libxpm4 php-bcmath php-gd php-ldap php-mbstring php-xml
  php7.4-bcmath php7.4-gd php7.4-ldap php7.4-mbstring php7.4-xml snmpd ttf-dejavu-cor
Suggested packages:
  libgd-tools libmyodbc odbc-postgresql tdsodbc unixodbc-bin lm-sensors snmp-mibs-dow
  snmptrapd zabbix-nginx-conf
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core fping libfontconfig1 libgd3 libjbig0 libjpeg-tu
  libmysqlclient21 libodbc1 libonig5 libopenipmi0 libsensors-config libsensors5 libsr
  libsnmp35 libtiff5 libwebp6 libxpm4 php-bcmath php-gd php-ldap php-mbstring php-xml
  php7.4-bcmath php7.4-gd php7.4-ldap php7.4-mbstring php7.4-xml snmpd ttf-dejavu-cor
  zabbix-agent zabbix-apache-conf zabbix-frontend-php zabbix-server-mysql
0 upgraded, 35 newly installed, 0 to remove and 73 not upgraded.
Need to get 10.8 MB of archives.
After this operation, 48.8 MB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal-updates/main amd64 libmysqlclient21 a
untu0.20.04.2 [1224 kB]
Get:2 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 zabbix-server-mysql a
focal [2473 kB]
8% [1 libmysqlclient21 738 kB/1224 kB 60%] [2 zabbix-server-mysql 312 kB/2473 kB 13%]
```

Si no se ve bien la orden de la captura anterior:

~$ sudo apt install -y zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent

Conectamos con el cliente de consola mysql y un usuario administrador para preparar la base de datos

```
javierrp@javierrp:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.22-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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

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

Creamos la base de datos atendiendo al juego de caracteres que requiere Zabbix en esta versión y creamos el usuario de la siguiente manera:

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

mysql> create user zabbix@localhost identified by 'practicas,ISE';
Query OK, 0 rows affected (0.05 sec)
```

Damos los permisos necesarios sobre la base de datos y salimos del cliente:

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

mysql> exit
Bye
javierrp@javierrp:~$
```

Ahora inicializamos una nueva base de datos a partir de un script SQL proporcionado por el paquete Zabbix con el usuario y contraseña recién creadas:

```
javierrp@javierrp:~$ zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql -u
ix
Enter password:

javierrp@javierrp:~$
javierrp@javierrp:~$ _
```

Por si no se ve la orden anterior:

~$ zcat /usr/share/doc/zabbix-server-mysql/create.sql.gz | mysql -u zabbix -p zabbix

Ahora vamos a configurar Zabbix Server para que utilice la contraseña de acceso a la base de datos:

```
javierrp@javierrp:~$ sudo nano /etc/zabbix/zabbix_server.conf
```

```
### Option: DBPassword
#       Database password.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
 DBPassword=practicas,ISE
```

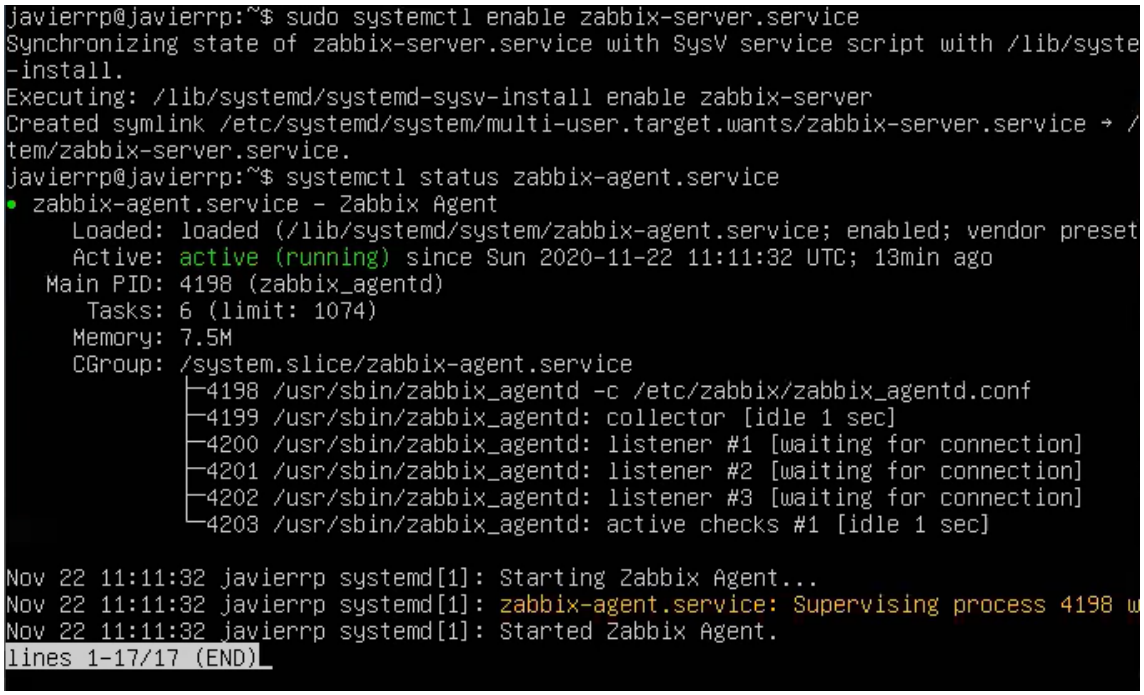
Ya si podemos arrancar el servicio zabbix-server:

```
javierrp@javierrp:~$ sudo systemctl start zabbix-server.service
javierrp@javierrp:~$ systemctl status zabbix-server.service
```

Y al hacer el ‘status’ deberíamos ver si todo está correcto algo como:

```
zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; disabled; vendor pres
   Active: active (running) since Sun 2020-11-22 11:23:30 UTC; 12s ago
  Process: 5403 ExecStart=/usr/sbin/zabbix_server -c $CONFFILE (code=exited, status
 Main PID: 5415 (zabbix_server)
    Tasks: 38 (limit: 1074)
   Memory: 74.6M
   CGroup: /system.slice/zabbix-server.service
           ├─5415 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
           ├─5418 /usr/sbin/zabbix_server: configuration syncer [synced configurati
           ├─5419 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes
           ├─5420 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0
           ├─5421 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000478
           ├─5422 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.00
           ├─5423 /usr/sbin/zabbix_server: history syncer #1 [processed 1 values, 1
           ├─5424 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0
           ├─5425 /usr/sbin/zabbix_server: history syncer #3 [processed 0 values, 0
           ├─5426 /usr/sbin/zabbix_server: history syncer #4 [processed 0 values, 0
           ├─5427 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in
           ├─5428 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 p
           ├─5429 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000
           ├─5430 /usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.0
           ├─5431 /usr/sbin/zabbix_server: poller #1 [got 0 values in 0.000065 sec,
           ├─5432 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000041 sec,
           ├─5433 /usr/sbin/zabbix_server: poller #3 [got 1 values in 0.000087 sec,
           ├─5434 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000064 sec,
           ├─5443 /usr/sbin/zabbix_server: poller #5 [got 0 values in 0.000065 sec,
           ├─5444 /usr/sbin/zabbix_server: unreachable poller #1 [got 0 values in 0
           ├─5445 /usr/sbin/zabbix_server: trapper #1 [processed data in 0.000000 s
           ├─5446 /usr/sbin/zabbix_server: trapper #2 [processed data in 0.000000 s
           ├─5447 /usr/sbin/zabbix_server: trapper #3 [processed data in 0.000000 s
           ├─5455 /usr/sbin/zabbix_server: trapper #4 [processed data in 0.000000 s
           ├─5456 /usr/sbin/zabbix_server: trapper #5 [processed data in 0.003791 s
           ├─5457 /usr/sbin/zabbix_server: icmp pinger #1 [got 0 values in 0.000068
           ├─5458 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alert
           ├─5459 /usr/sbin/zabbix_server: alerter #1 started
```

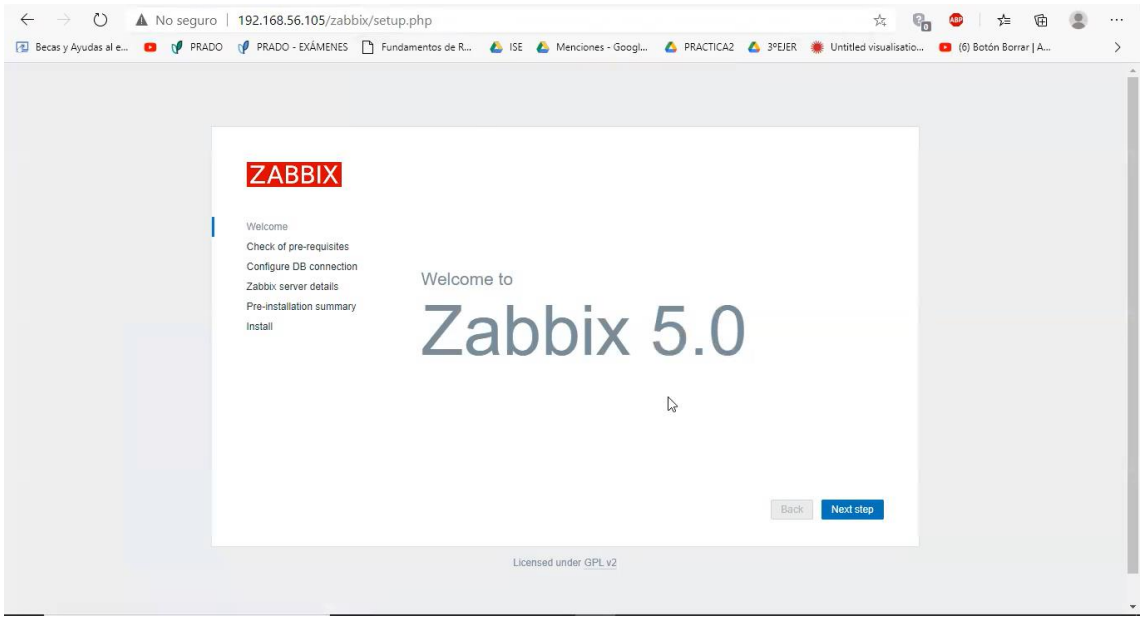
Y para que el servicio Zabbix Server inicie automáticamente con cada arranque de Ubuntu Server lo habilitamos tal que:



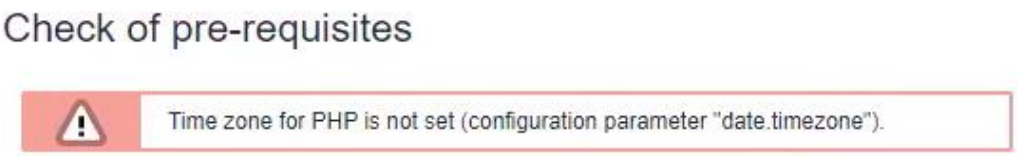
Para que el agente Zabbix también inicie automáticamente con cada arranque y reiniciamos el servicio de apache2 para poder conectarnos desde el buscador correctamente a través de la IP y poder configurar el frontend:

Frontend

Este se puede configurar desde cualquier buscador, por lo que yo uso el de Windows introduciendo '192.168.56.105/zabbix':



Al hacer next step, saldrán los requisitos mínimos que necesitas. Debería estar todo correcto, pero a mí personalmente me salió un error del "date.timezone", así que paso a explicar cómo lo arreglé por si a alguien le pasa.



Abrimos como superusuarios el siguiente archivo de configuración:

```
root@javierrp:/home/javierrp# vi /etc/php/7.4/apache2/php.ini
```

Y modificamos el siguiente valor (el ; de delante es como comentarlo para que venga con valores por defecto, por lo que hay que quitarlos):

date.timezone =Europe/Madrid

```
;extension=sodium
;extension=sqlite3
;extension=tidy
;extension=xmlrpc
;extension=xsl

;;;;;;;;;;;;;;;;;;;
; Module Settings ;
;;;;;;;;;;;;;;;;;;;

[CLI Server]
; Whether the CLI web server uses ANSI color coding in its terminal output.
cli_server.color = On

[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone =Europe/Madrid

; http://php.net/date.default-latitude
;date.default_latitude = 31.7667

; http://php.net/date.default-longitude
;date.default_longitude = 35.2333

; http://php.net/date.sunrise-zenith
;date.sunrise_zenith = 90.583333

; http://php.net/date.sunset-zenith
;date.sunset_zenith = 90.583333

[filter]
; http://php.net/filter.default
;filter.default = unsafe_raw

; http://php.net/filter.default-flags
-- INSERT --                                                    962
```

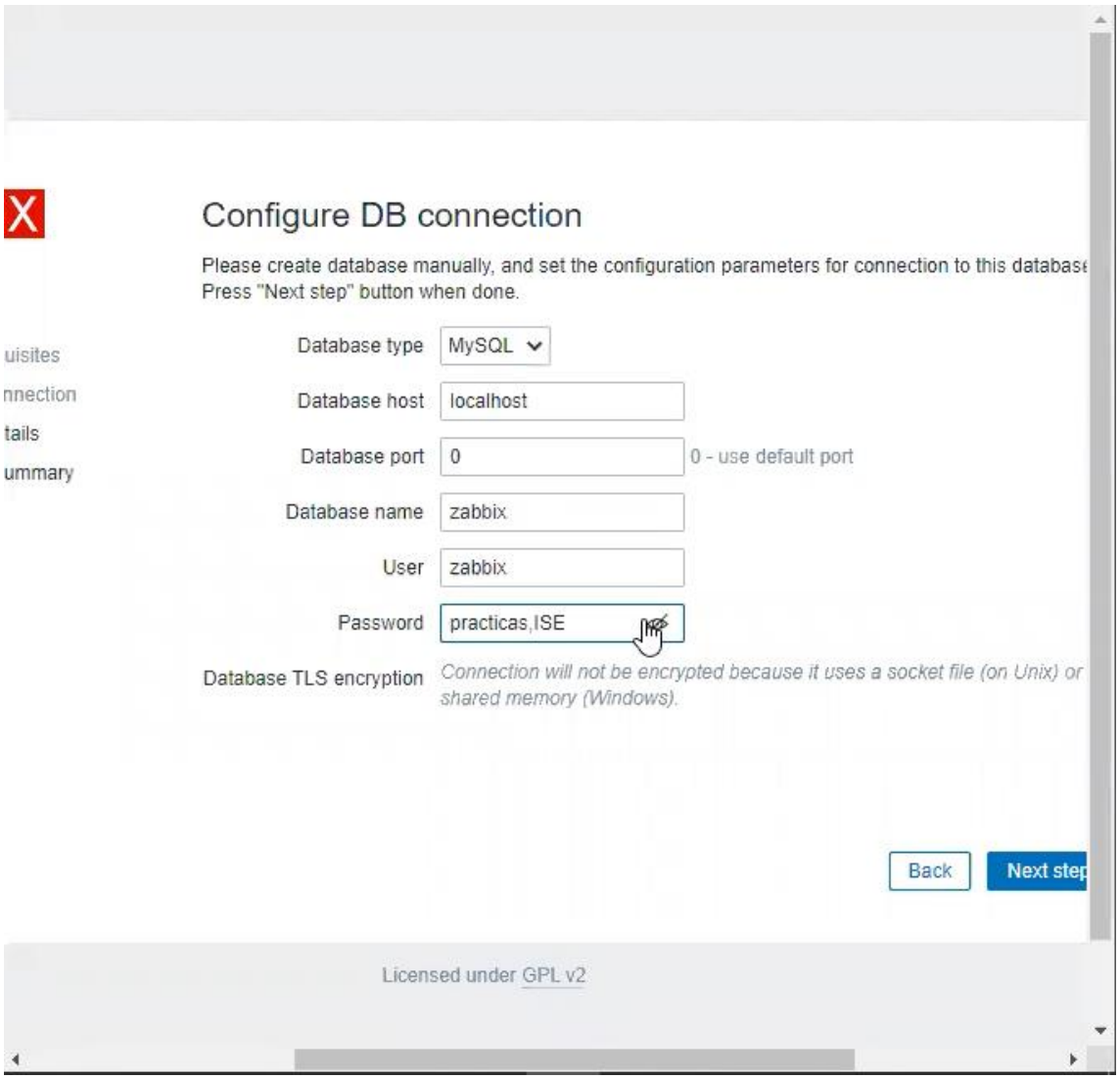
Se reinicia el sistema de apache2:

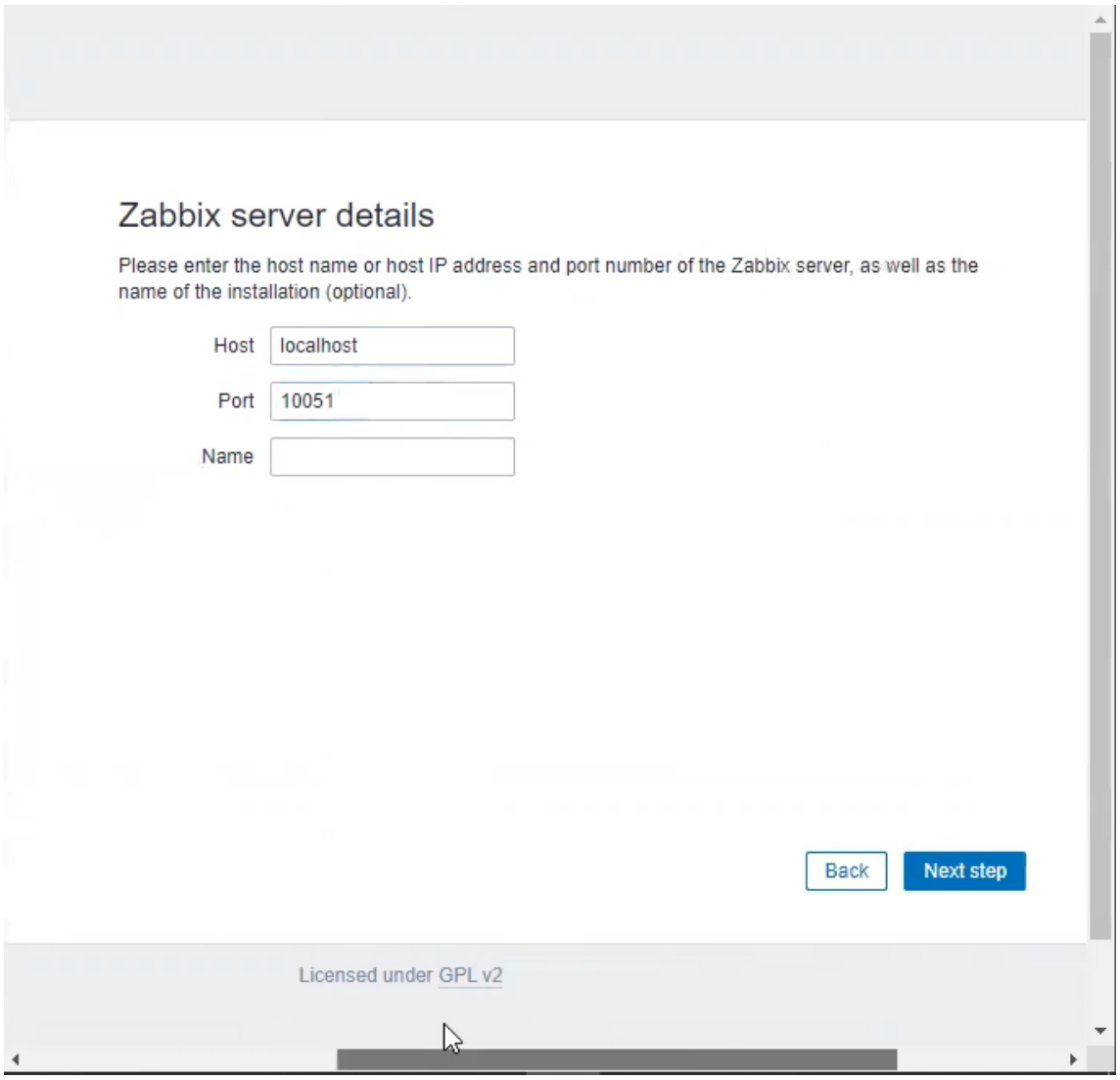
```
root@javierrp:/home/javierrp# systemctl restart apache2
root@javierrp:/home/javierrp# sudo systemctl restart zabbix-server
root@javierrp:/home/javierrp# sudo systemctl restart zabbix-agent
```

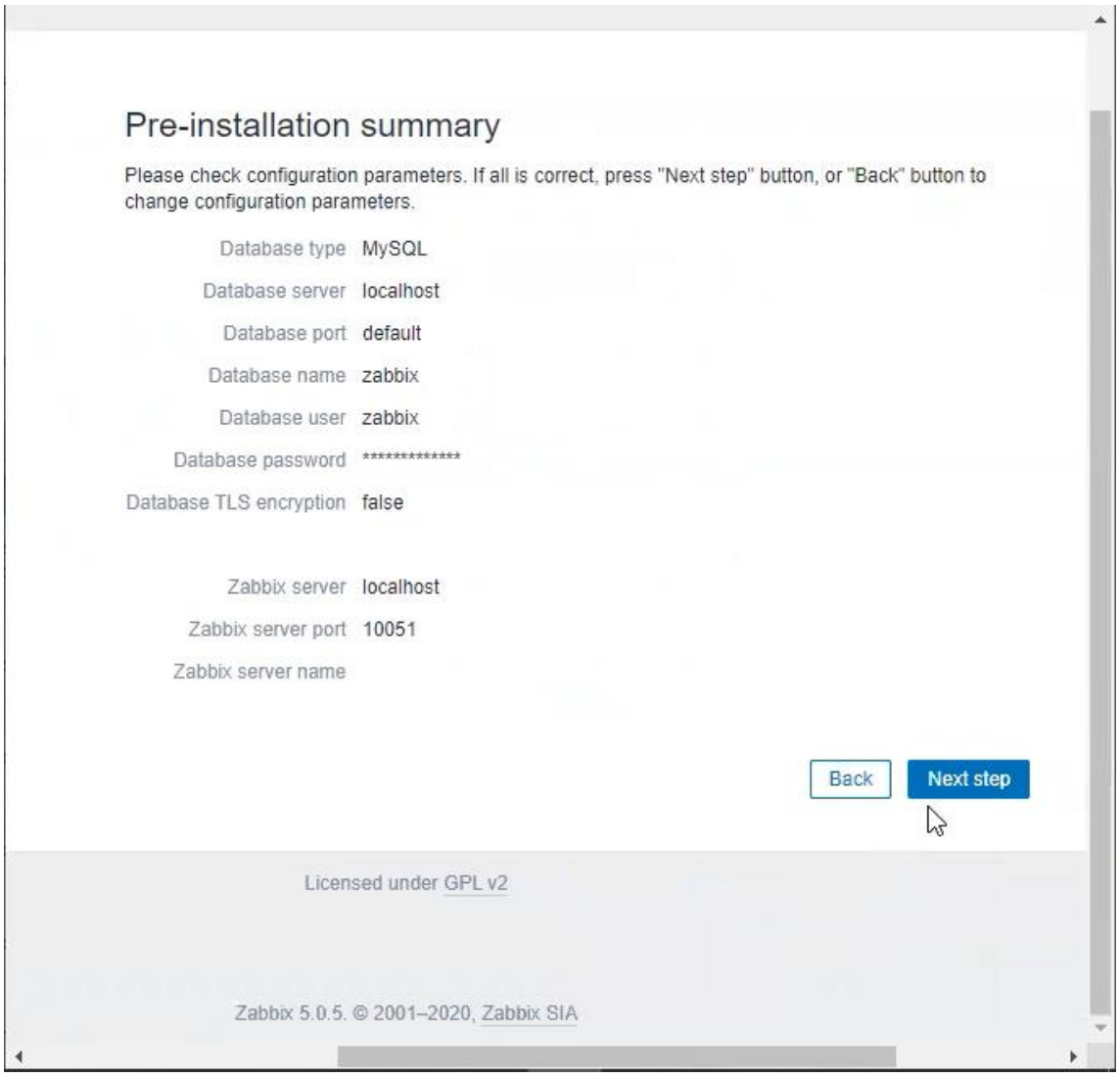
Este paso es para prevenir, como sabemos que Zabbix usará el puerto 10050, permitimos desde ya con el cortafuegos que Ubuntu oiga por ese puerto.

```
root@javierrp:/home/javierrp# sudo ufw allow 10050/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
root@javierrp:/home/javierrp#
```

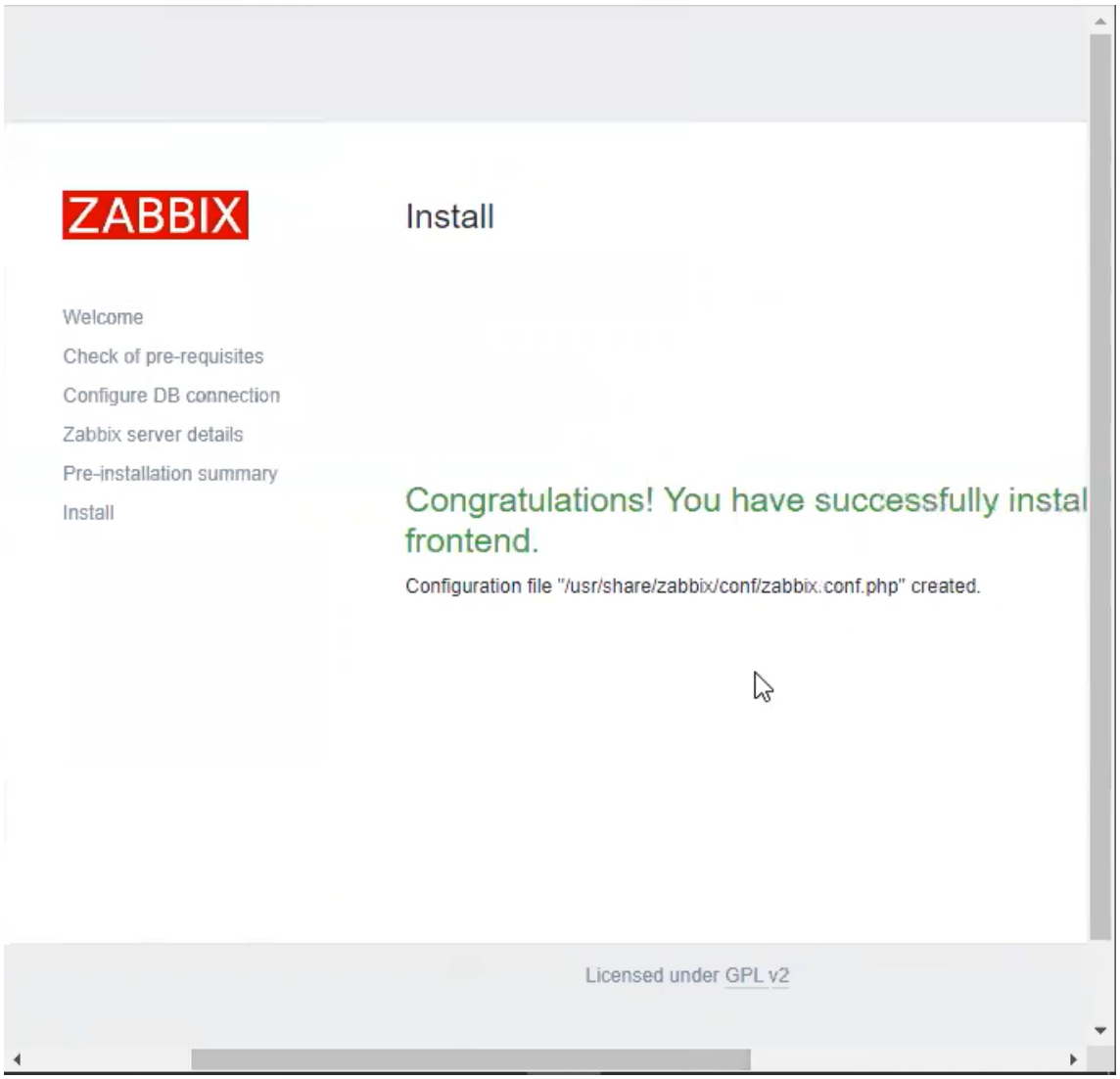
El fallo de timezone habrá desaparecido y continuamos configurando el Frontend:




Zabbix server details
Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).
Host
localhost
Port
10051
Name
Back
Next step
Licensed under GPL v2

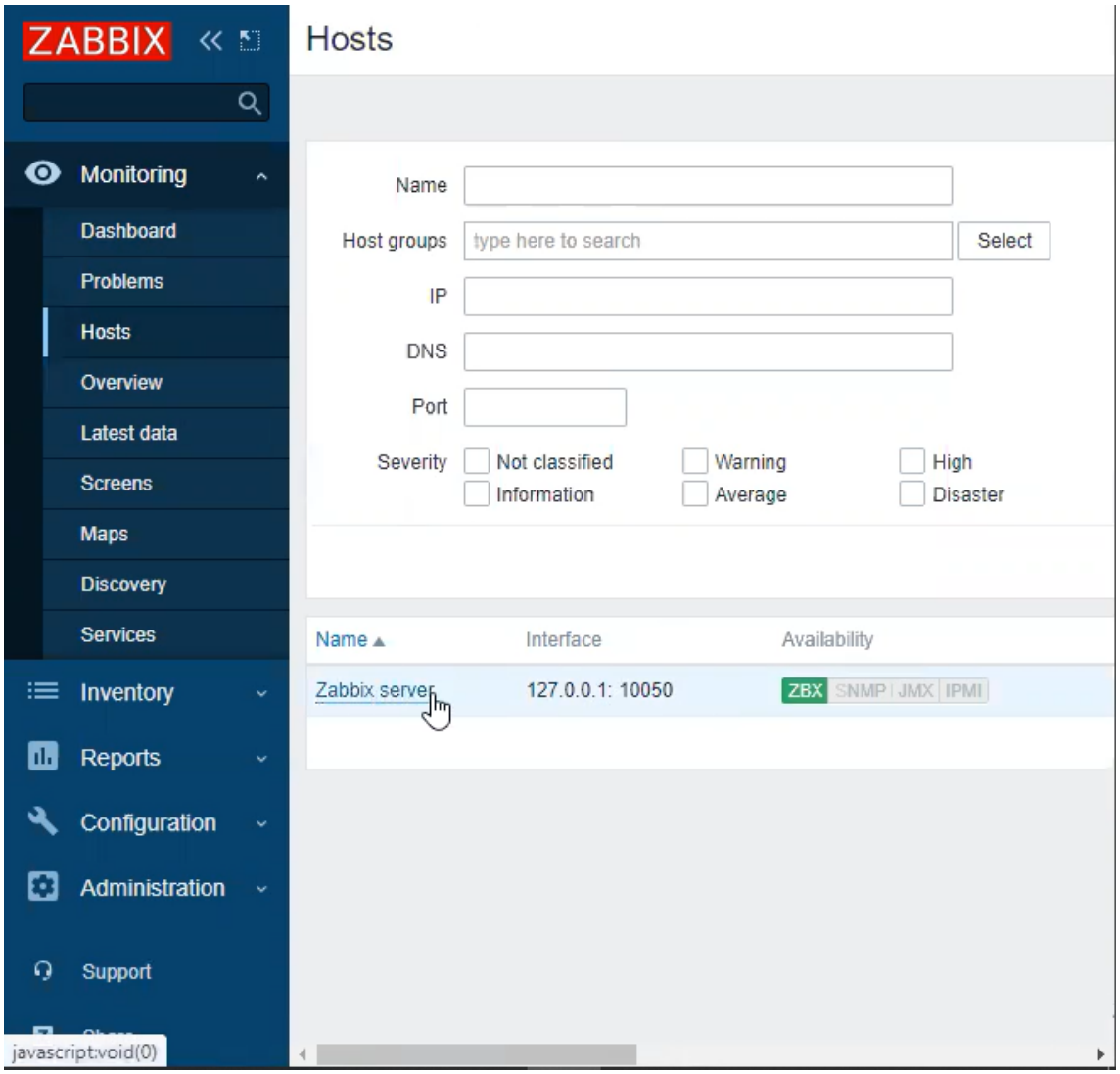

Pre-installation summary
Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.
Database type MySQL
Database server localhost
Database port default
Database name zabbix
Database user zabbix
Database password **********
Database TLS encryption false
Zabbix server localhost
Zabbix server port 10051
Zabbix server name
Back
Next step
Licensed under GPL v2
Zabbix 5.0.5. © 2001–2020, Zabbix SIA

Esta es la pantalla tras seguir todos los pasos que te saldrá si ha ido correctamente el proceso:



Iniciamos sesión con el usuario y contraseña por defecto: Admin y zabbix, respectivamente.

Y con eso ya entramos a ver las máquinas que tenemos monitorizadas.



En este caso, Zabbix server sería la máquina de Ubuntu Server, pero vamos a crearla de nuevo para ver el procedimiento y para tener un nombre descriptivo con una correcta IP.

Configuración para ser monitorizado

En Ubuntu Server, con todos los paquetes y programas ya instalados para el host, simplemente modificamos el siguiente archivo de configuración de zabbix:

```
root@javierrp:/home/javierrp# sudo vi /etc/zabbix/zabbix_agentd.conf
```

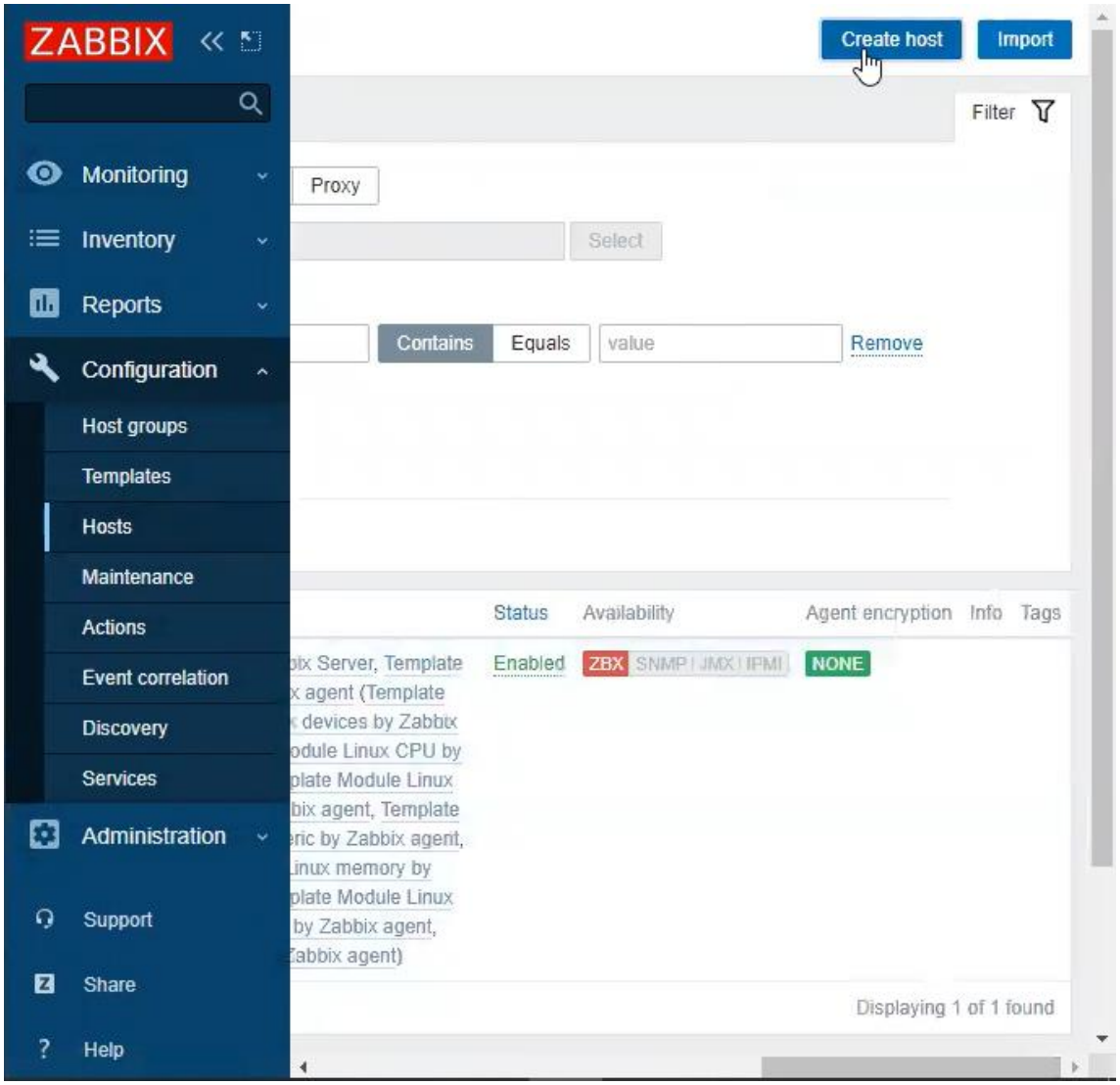
```
## Option: Server
#       List of comma delimited IP addresses, optionally in CIDR notation, or D
rvers and Zabbix proxies.
#       Incoming connections will be accepted only from the hosts listed here.
#       If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127
qually
#       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.examp
#
# 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

### Option: ListenIP
#       List of comma delimited IP addresses that the agent should listen on.
#       First IP address is sent to Zabbix server if connecting to it to retrie
cks.
#
```

Y reiniciamos zabbix-agent, dándole permisos con el cortafuegos:

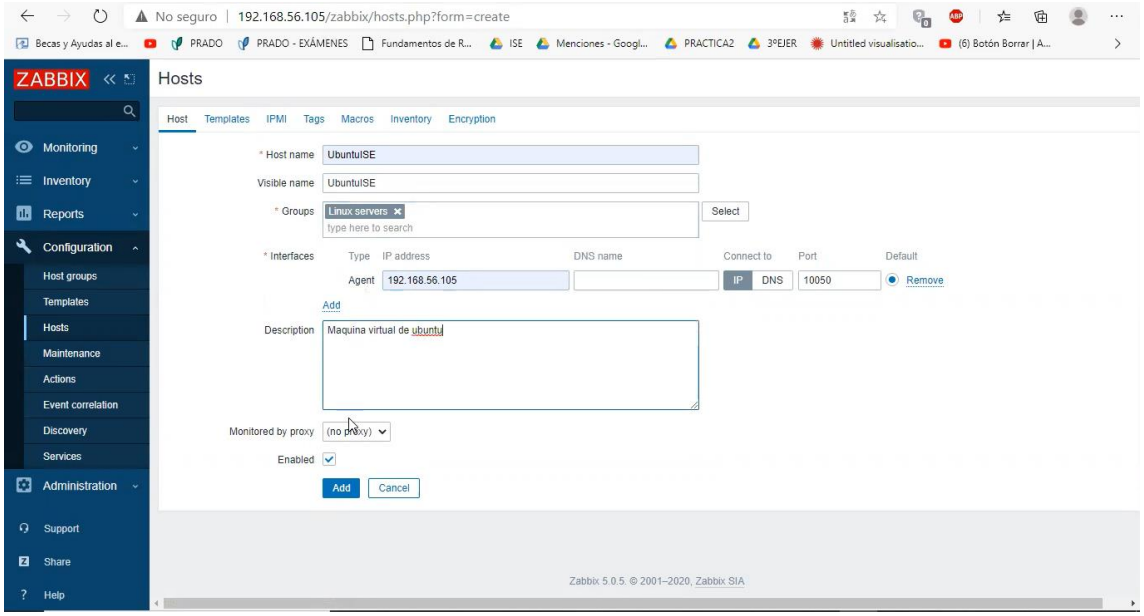
```
root@javierrp:/home/javierrp# sudo systemctl restart zabbix-agent
root@javierrp:/home/javierrp# sudo ufw allow zabbix-agent
Rule added
Rule added (v6)
root@javierrp:/home/javierrp# _
```

Añadir Ubuntu como monitorizado

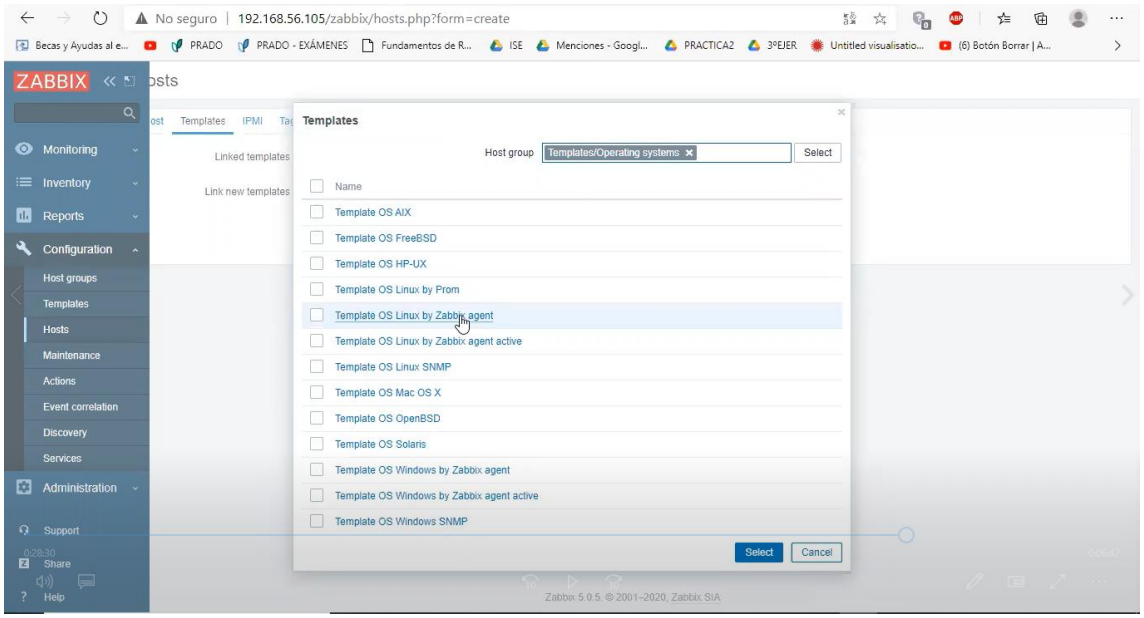
Ahora vamos a añadir una nueva máquina monitorizada. Para ello vamos al menú lateral izquierdo en *Configuration -> hosts -> create host (*Este último se encuentra arriba a la derecha*)*

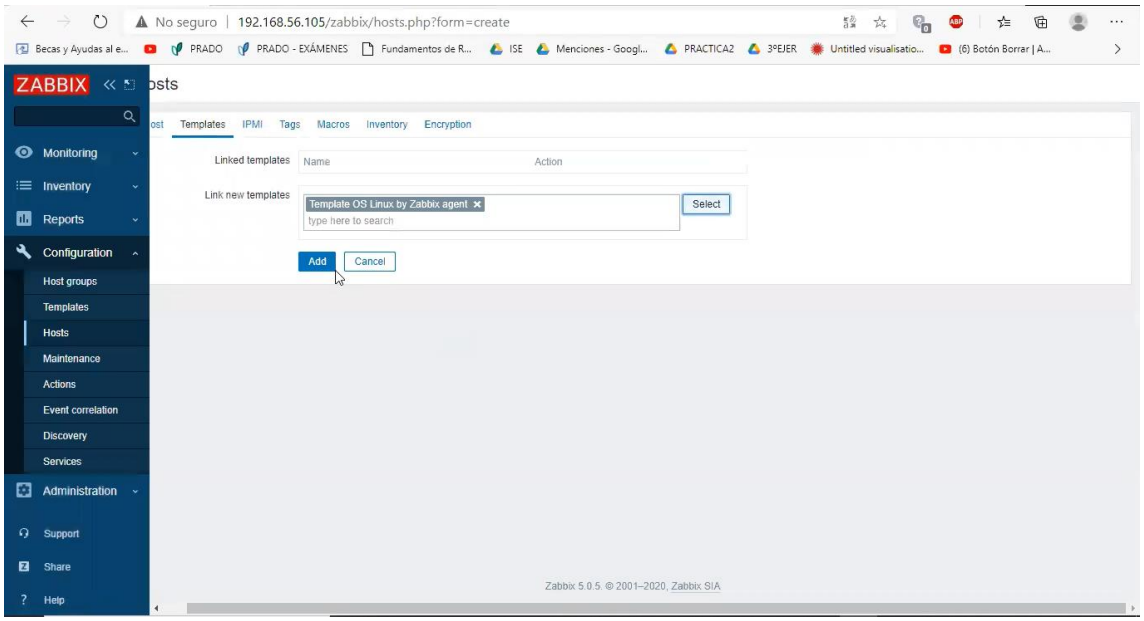


Configuramos los nombres y grupos de servidores a los que pertenecerá.



Y antes de dar en “Add”, vamos a la solapa superior donde pone “Templates” y añadimos el siguiente:





Ya se habrá creado y podemos verlo en nuestro *Monitoring -> Hosts*

Instalación en centos

Añadimos el repositorio correspondiente

```
[root@localhost ~]# rpm -ivh http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/zabbix-release-3.4-2.el
7.noarch.rpm
Recuperando http://repo.zabbix.com/zabbix/3.4/rhel/7/x86_64/zabbix-release-3.4-2.el7.noarch.rpm
advertencia:/var/tmp/rpm-tmp.woO1fT: EncabezadoV4 RSA/SHA512 Signature, ID de clave a14fe591: NOKEY
Verifying...                          ################################# [100%]
Preparando...                         ################################# [100%]
Actualizando / instalando...
   1:zabbix-release-3.4-2.el7         ################################# [100%]
[root@localhost ~]#
```

Como centos solo va a ser monitorizado y no será host, instalamos solamente el agente.

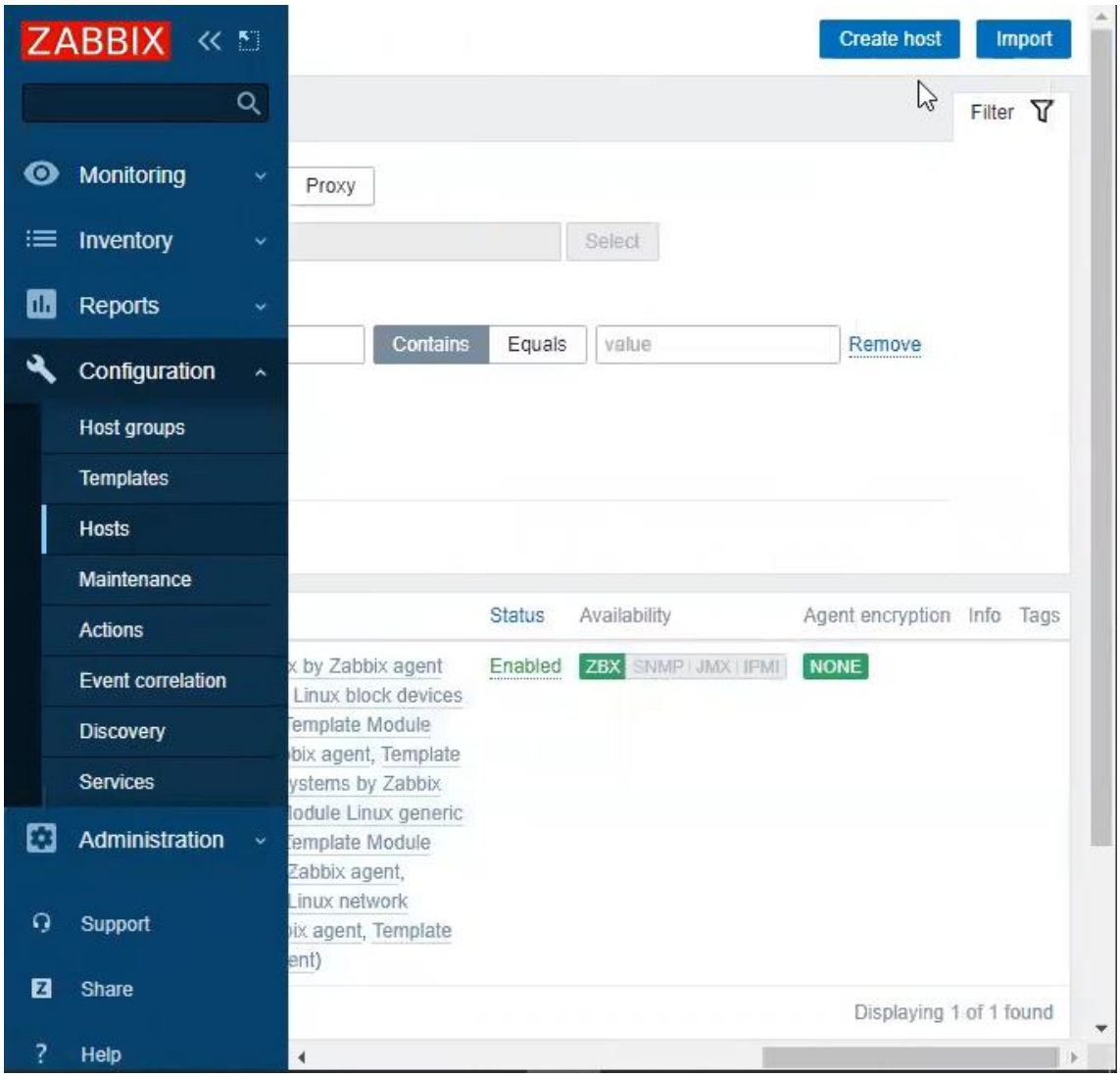
```
[root@localhost ~]# yum install zabbix-agent
Zabbix Official Repository - x86_64                              96 kB/s | 141 kB
Zabbix Official Repository non-supported - x86_64               2.7 kB/s | 2.3 kB
Dependencias resueltas.
================================================================================
 Paquete                 Arquitectura      Versión                  Repositorio
================================================================================
Instalando:
 zabbix-agent            x86_64            3.4.15-1.el7             zabbix
Instalando dependencias:
 compat-openssl10        x86_64            1:1.0.2o-3.el8           AppStream
 make                    x86_64            1:4.2.1-10.el8           BaseOS

Resumen de la transacción
================================================================================
Instalar  3 Paquetes

Tamaño total de la descarga: 2.0 M
Tamaño instalado: 5.9 M
¿Está de acuerdo [s/N]?: _
```

Ya podemos iniciarlo y, como sabemos que tenemos el cortafuegos activado, añadimos el puerto de Zabbix a la configuración.

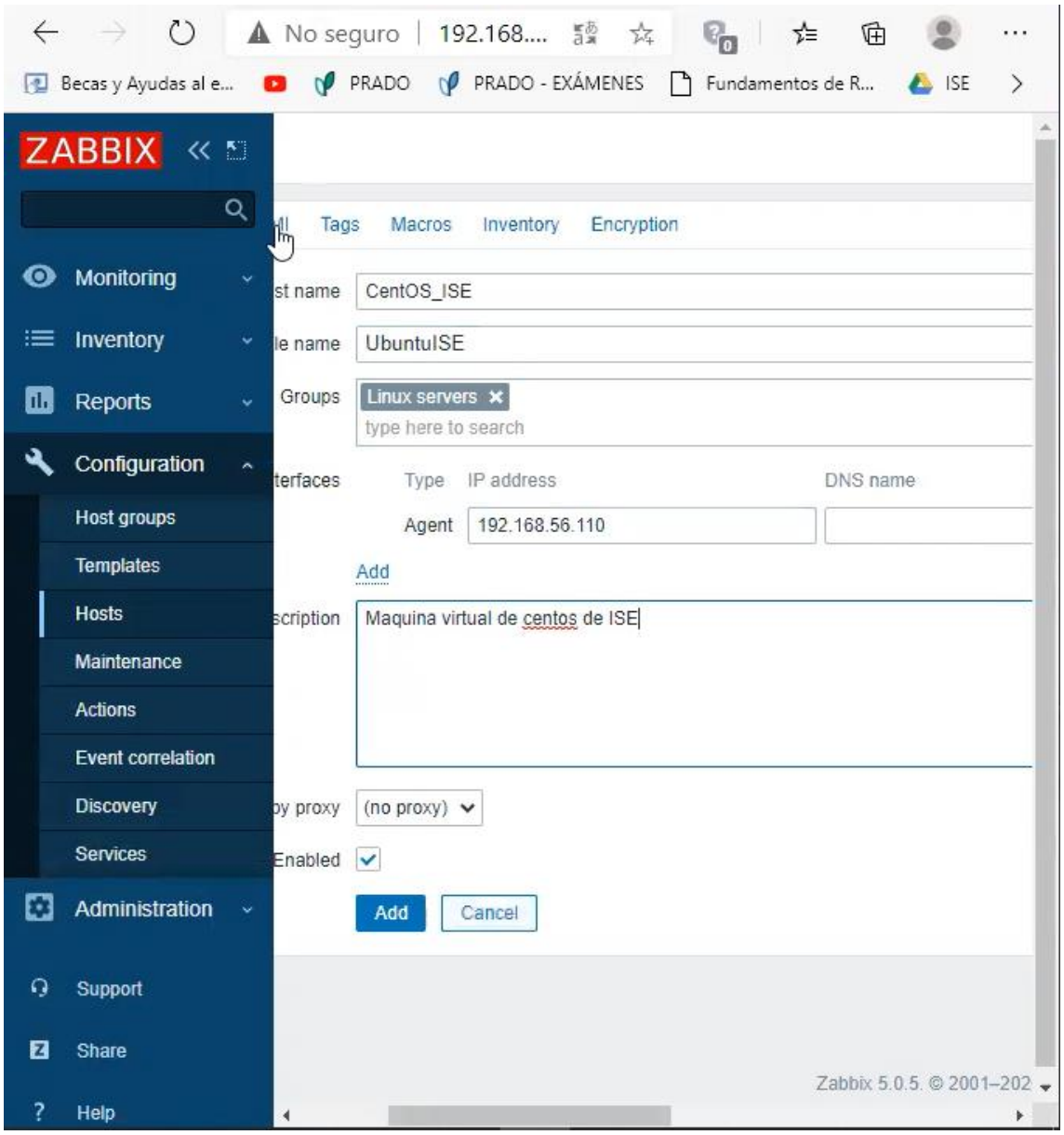
```
[root@localhost ~]# service zabbix-agent start
Redirecting to /bin/systemctl start zabbix-agent.service
[root@localhost ~]# firewall-cmd --add-port=10050/tcp --permanent
success
[root@localhost ~]# firewall-cmd --reload
success
[root@localhost ~]#
```

Añadir la máquina de centOS en Zabbix

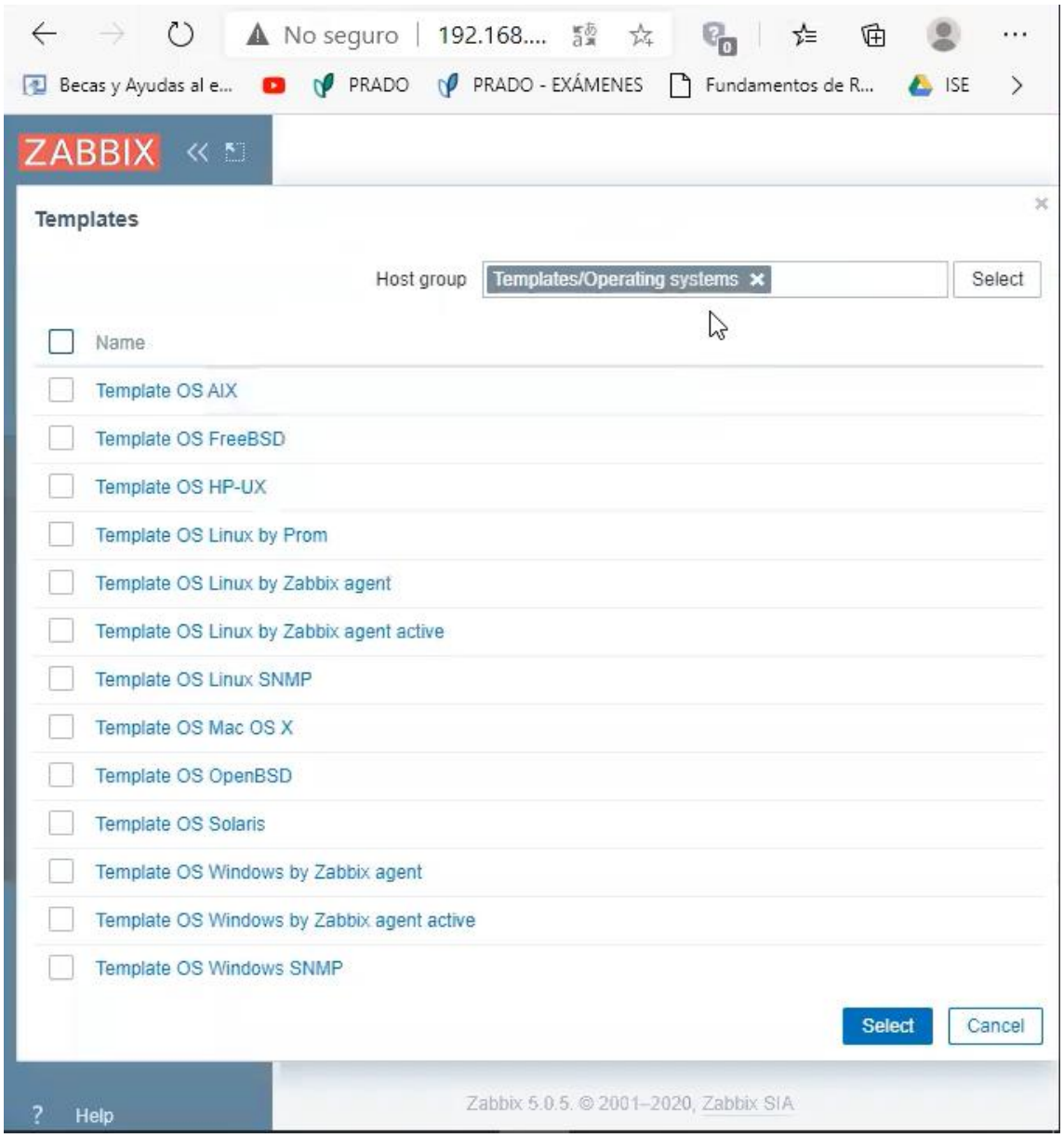
Volvemos a crear un nuevo host



Le añadimos el nombre, grupo, IP y descripción correctamente:



Configuramos los “Templates” antes de darle a “Add” de nuevo



Para que el agente Zabbix inicie automáticamente con cada arranque

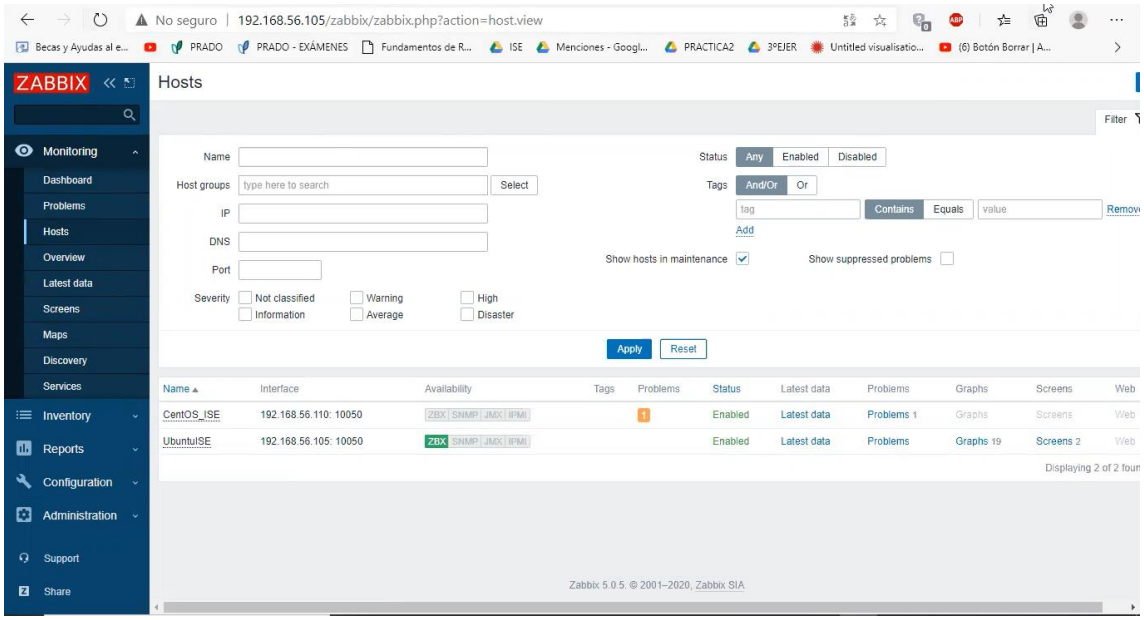
```
[root@localhost ~]# systemctl enable zabbix-agent
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /u
system/zabbix-agent.service.
[root@localhost ~]# systemctl start zabbix-agent
[root@localhost ~]#
```

Y configuramos lo mismo que tuvimos que hacer en Ubuntu Server:

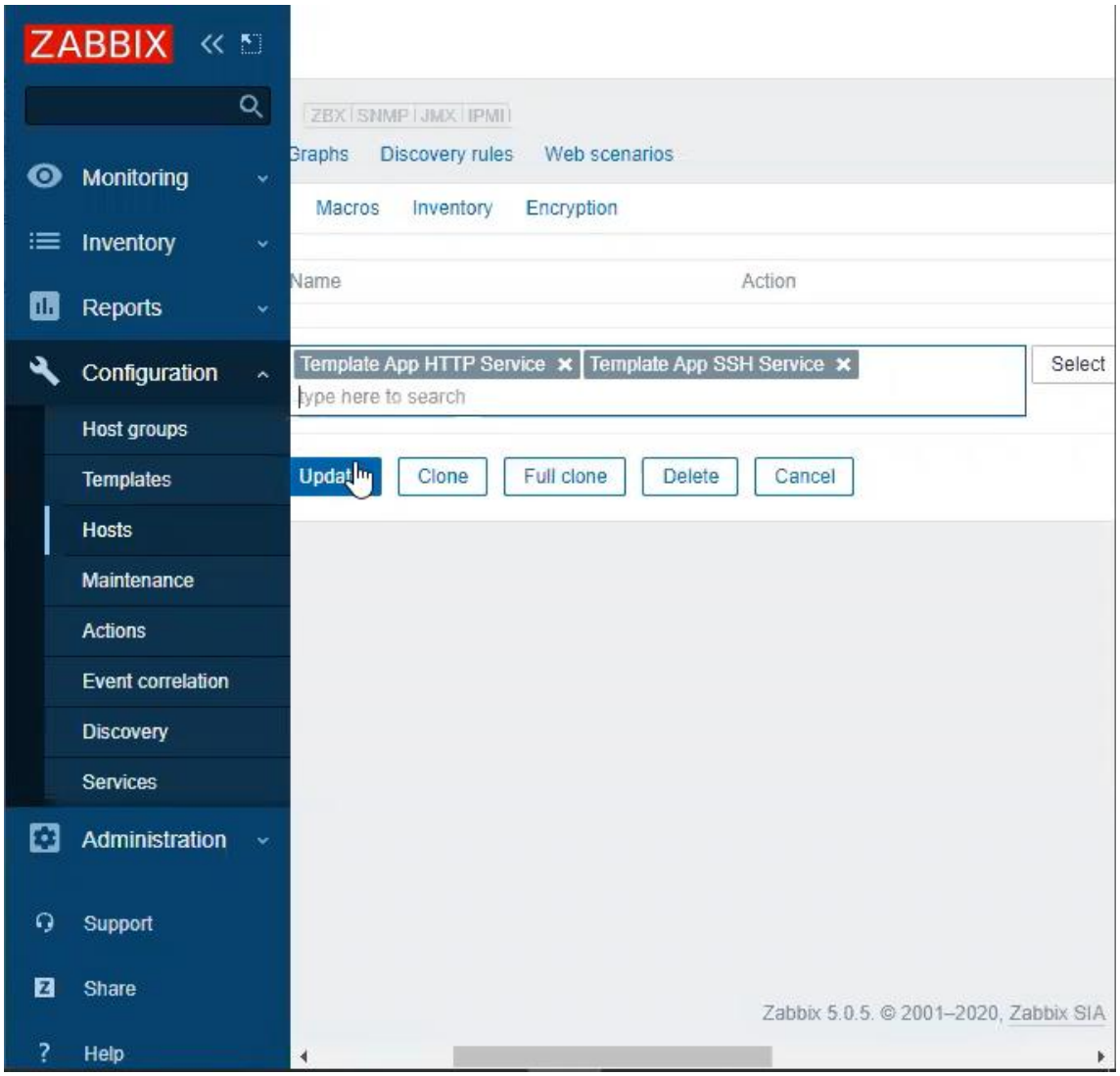
```
[root@localhost ~]# vi /etc/zabbix/zabbix_agentd.conf
```

```
### Option: Server
#       List of comma delimited IP addresses, optionally in CIDR notation, or hostnam
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
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.domain
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.56.105_
```

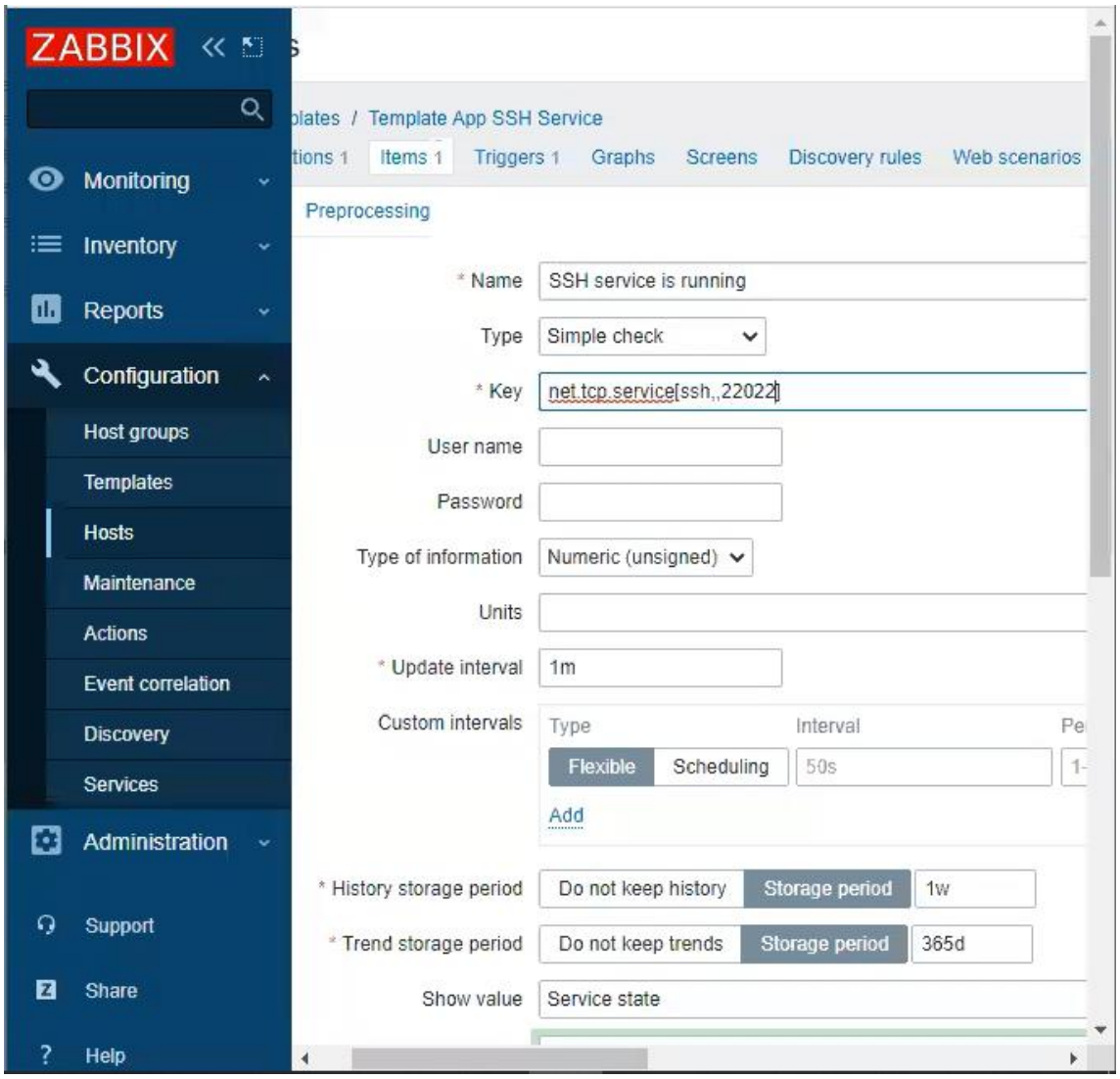
Y con eso ya vemos como están ambas creadas correctamente:



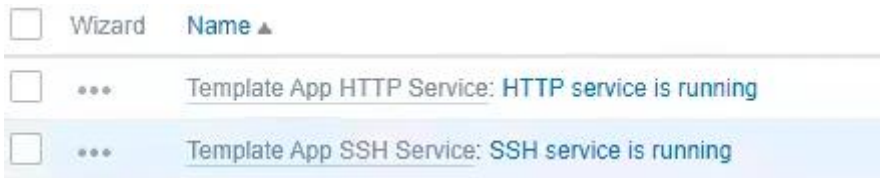
Hay varias formas de añadir los templates necesarios para el control de SSH y HTTP. El que yo he realizado es yendo a la configuración de una de las máquinas (*Configuration->Hosts-> Centos->Tempates*) y añadiendo los templates necesarios



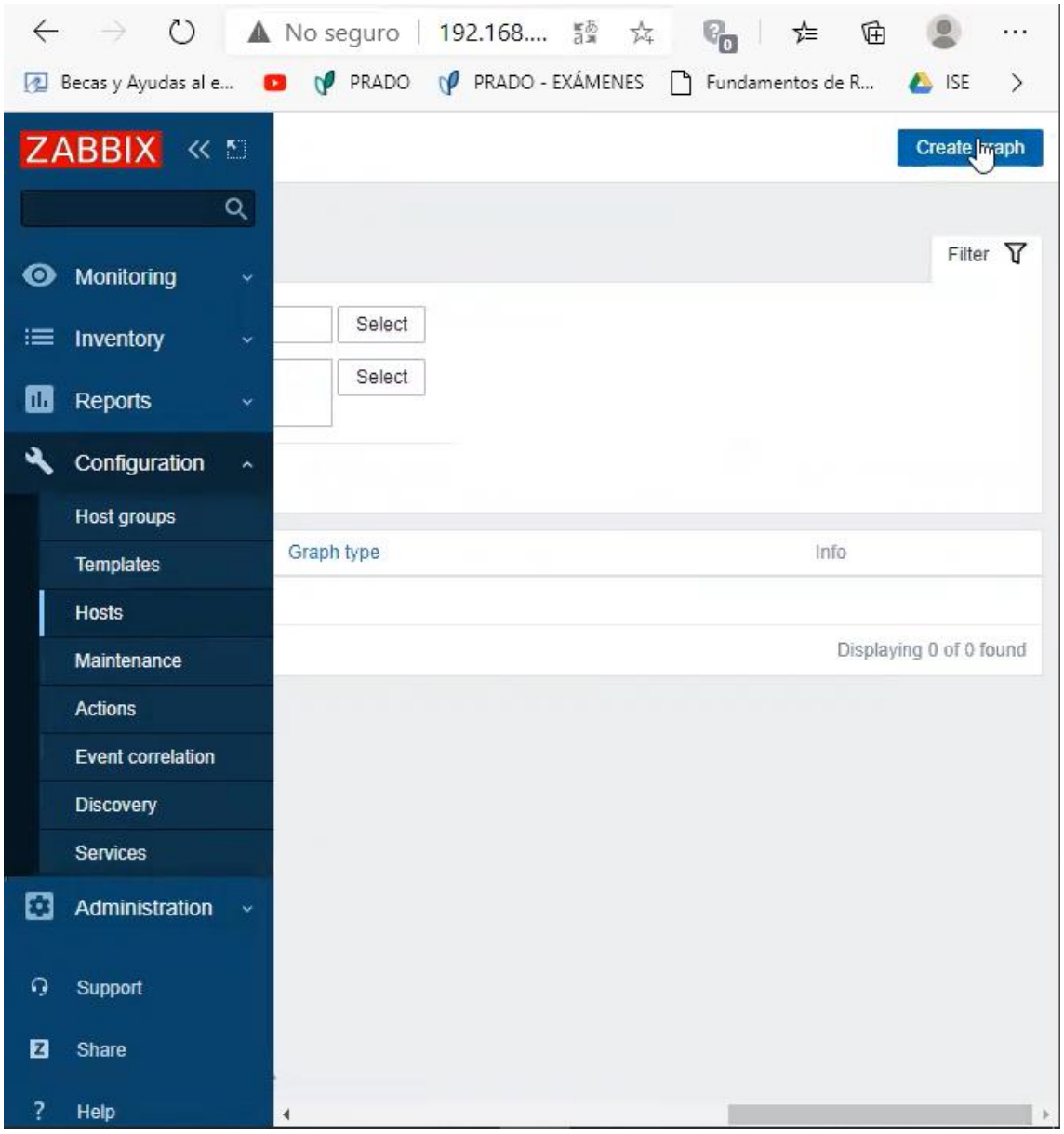
Y creo también los Items (de cada uno de los dos templates recién añadidos) que harán falta para las gráficas de después:

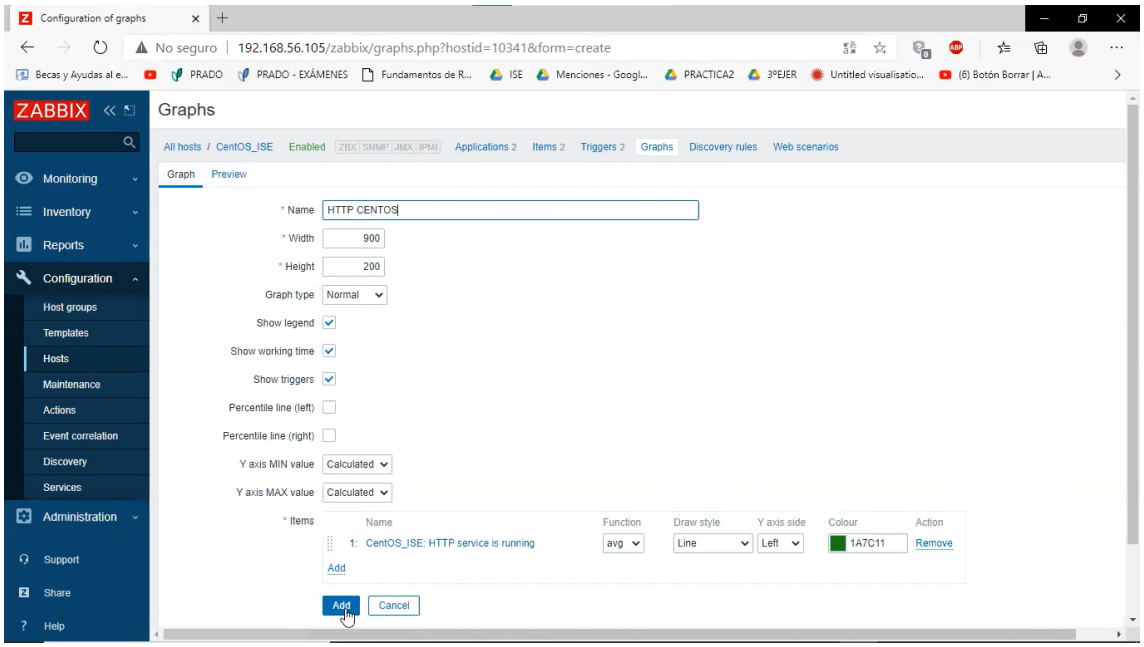


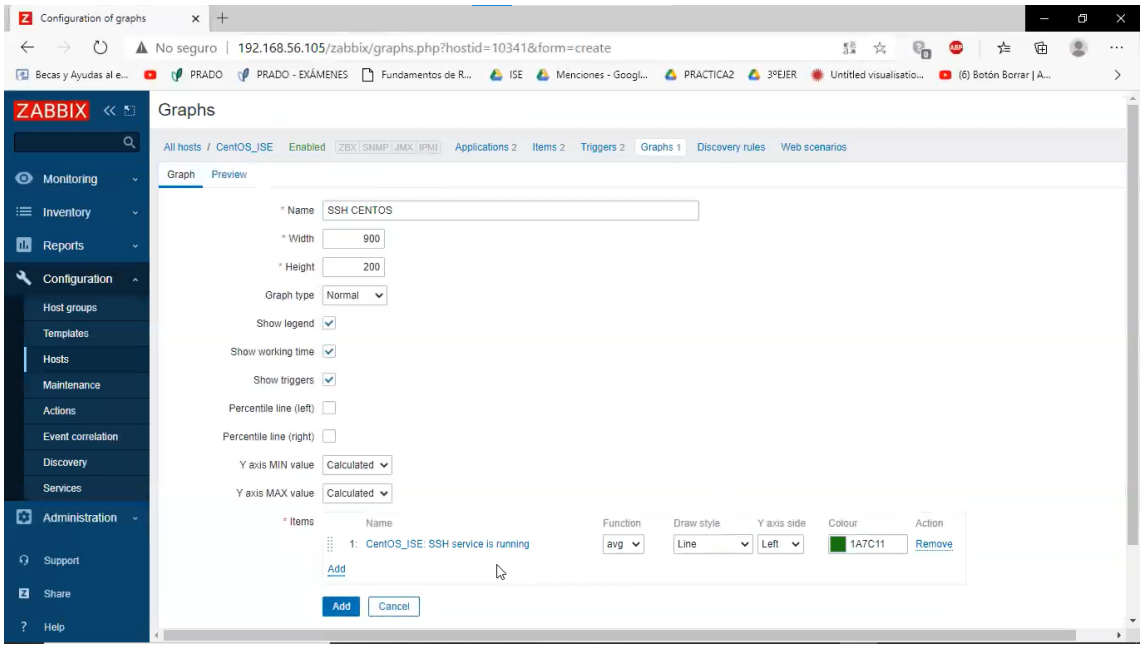
Quedando así configurado:

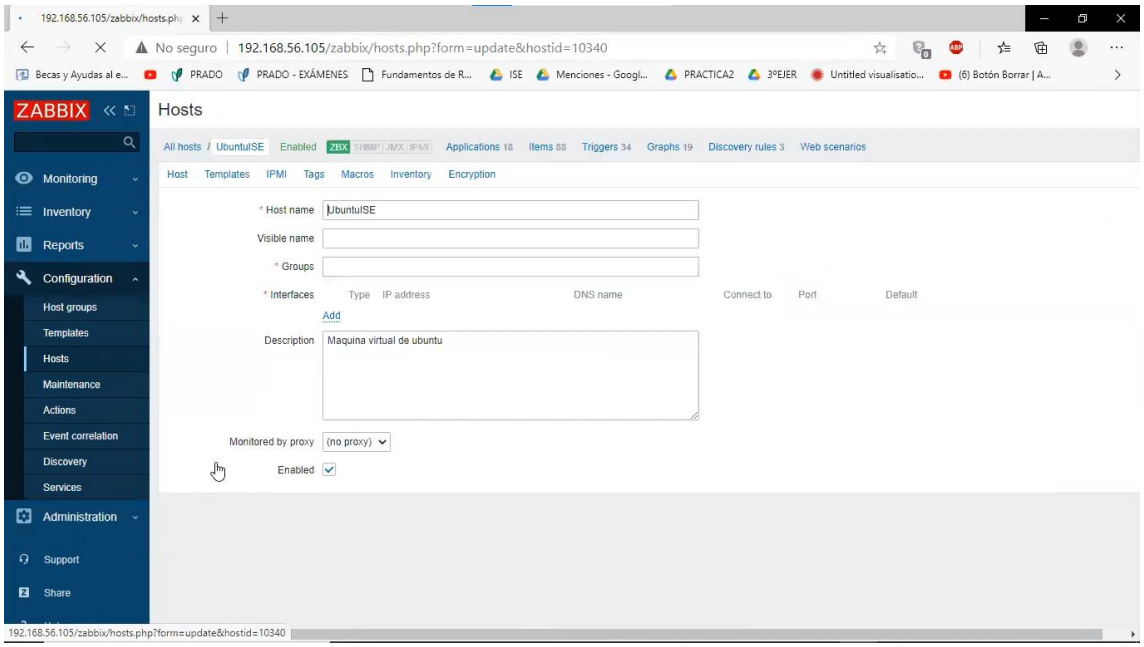


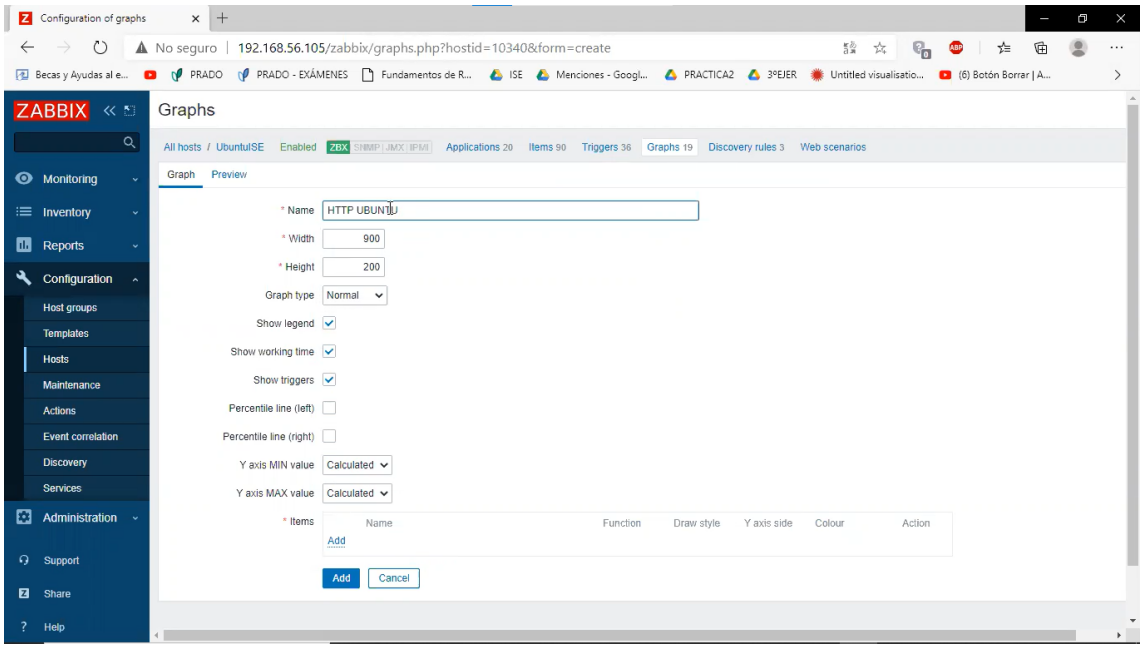
Vuelvo a ir a *Configuration->Hosts->centos->graphs->create graph* y *Configuration->Hosts->ubuntu->graphs->create graph* para crear las gráficas que muestren el funcionamiento de ssh y http en cada una (4 gráficas en total):

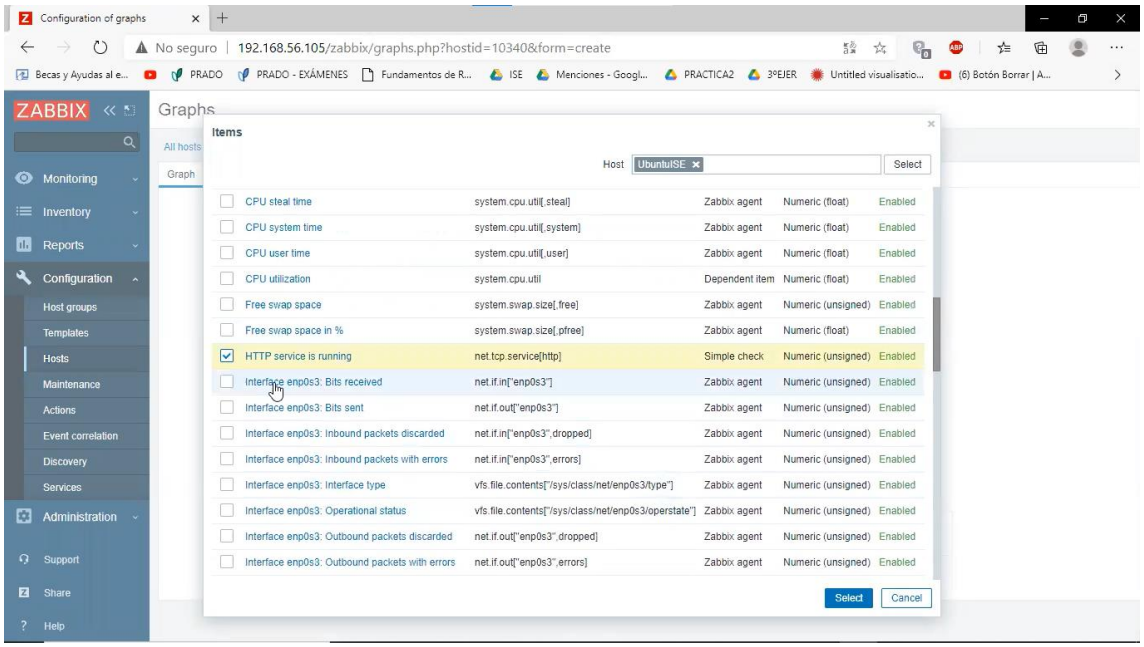


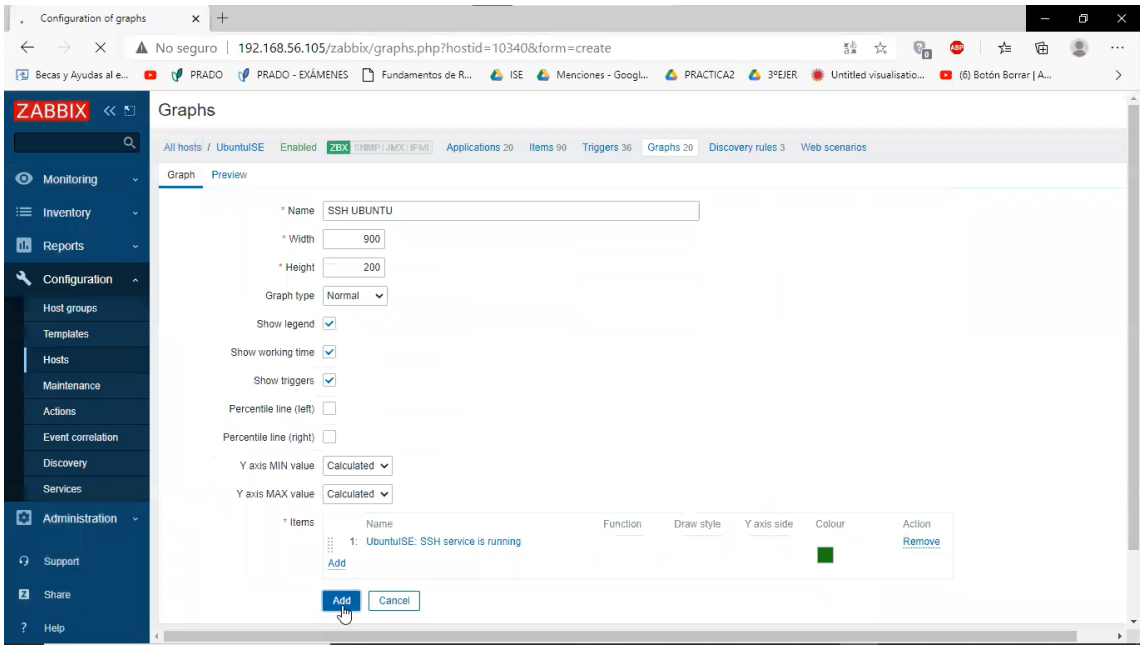

Configuration of graphs
No seguro | 192.168.56.105/zabbix/graphs.php?hostid=10341&form=create
ZABBIX
Graphs
All hosts / CentOS_ISE Enabled ZBX SNMP JMX IPMI Applications 2 Items 2 Triggers 2 Graphs Discovery rules Web scenarios
Graph Preview
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help
* Name HTTP CENTOS
* Width 900
* Height 200
Graph type Normal
Show legend
Show working time
Show triggers
Percentile line (left)
Percentile line (right)
Y axis MIN value Calculated
Y axis MAX value Calculated
* Items
Name Function Draw style Y axis side Colour Action
1: CentOS_ISE: HTTP service is running avg Line Left 1A7C11 Remove
Add
Add Cancel


Configuration of graphs
No seguro | 192.168.56.105/zabbix/graphs.php?hostid=10341&form=create
ZABBIX
Graphs
All hosts / CentOS_ISE Enabled ZBX SNMP JMX IPMI Applications 2 Items 2 Triggers 2 Graphs 1 Discovery rules Web scenarios
Graph Preview
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help
* Name SSH CENTOS
* Width 900
* Height 200
Graph type Normal
Show legend
Show working time
Show triggers
Percentile line (left)
Percentile line (right)
Y axis MIN value Calculated
Y axis MAX value Calculated
* Items
Name Function Draw style Y axis side Colour Action
1: CentOS_ISE: SSH service is running avg Line Left 1A7C11 Remove
Add
Add Cancel


192.168.56.105/zabbix/hosts.php
No seguro | 192.168.56.105/zabbix/hosts.php?form=update&hostid=10340
ZABBIX
Hosts
All hosts / UbuntuISE Enabled ZBX SNMP JMX IPMI Applications 18 Items 88 Triggers 34 Graphs 19 Discovery rules 3 Web scenarios
Host Templates IPMI Tags Macros Inventory Encryption
* Host name UbuntuISE
Visible name
* Groups
* Interfaces Type IP address DNS name Connect to Port Default
Add
Description Maquina virtual de ubuntu
Monitored by proxy (no proxy)
Enabled
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
192.168.56.105/zabbix/hosts.php?form=update&hostid=10340

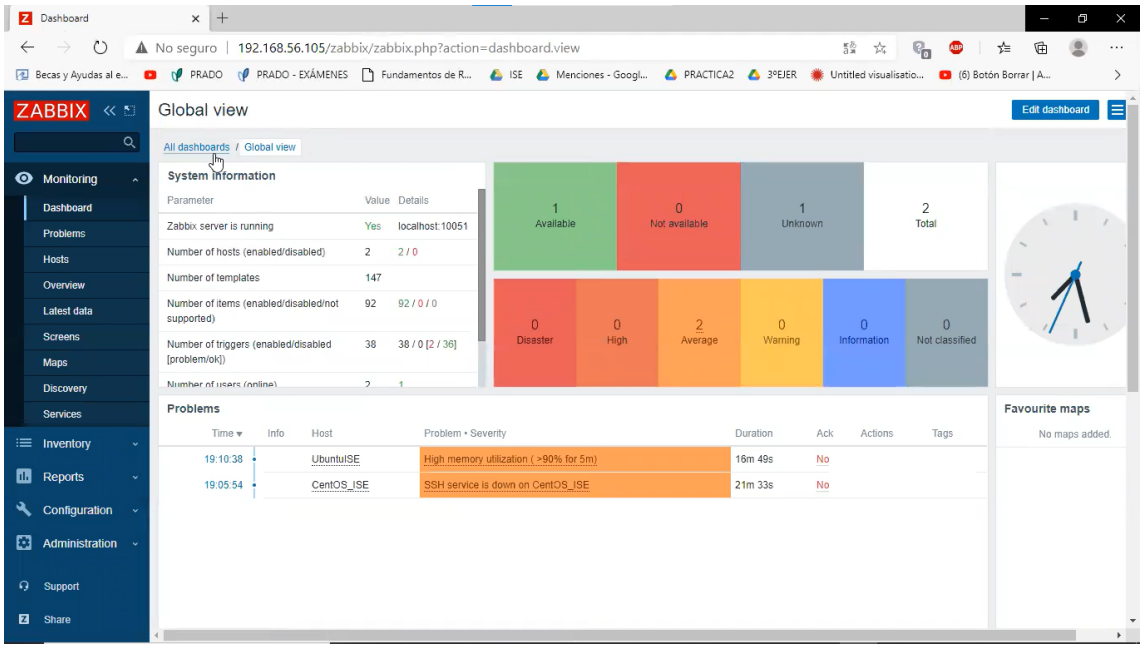

Configuration of graphs
No seguro | 192.168.56.105/zabbix/graphs.php?hostid=10340&form=create
ZABBIX
Graphs
All hosts / UbuntuISE Enabled ZBX SNMP JMX IPMI Applications 20 Items 90 Triggers 36 Graphs 19 Discovery rules 3 Web scenarios
Graph Preview
* Name HTTP UBUNTU
* Width 900
* Height 200
Graph type Normal
Show legend
Show working time
Show triggers
Percentile line (left)
Percentile line (right)
Y axis MIN value Calculated
Y axis MAX value Calculated
* Items Name Function Draw style Y axis side Colour Action
Add
Add Cancel
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help

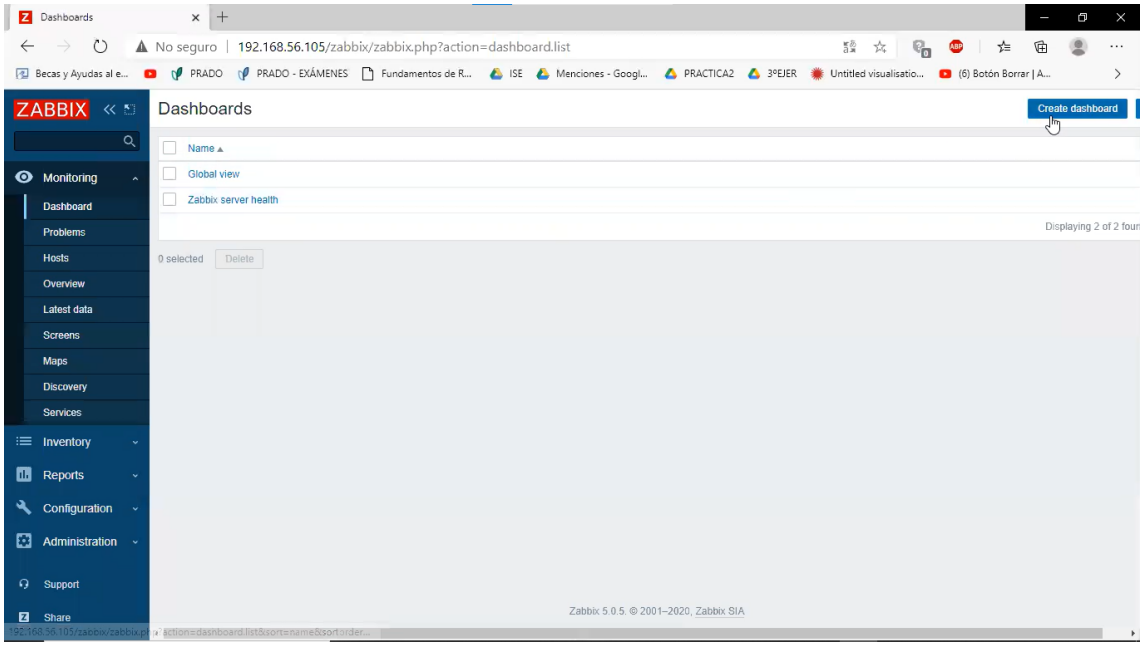

Configuration of graphs
No seguro | 192.168.56.105/zabbix/graphs.php?hostid=10340&form=create
ZABBIX
Graphs
Items
Host UbuntuISE
Select
CPU steal time system.cpu.util[,steal] Zabbix agent Numeric (float) Enabled
CPU system time system.cpu.util[,system] Zabbix agent Numeric (float) Enabled
CPU user time system.cpu.util[,user] Zabbix agent Numeric (float) Enabled
CPU utilization system.cpu.util Dependent item Numeric (float) Enabled
Free swap space system.swap.size[,free] Zabbix agent Numeric (unsigned) Enabled
Free swap space in % system.swap.size[,pfree] Zabbix agent Numeric (float) Enabled
HTTP service is running net.tcp.service[http] Simple check Numeric (unsigned) Enabled
Interface enp0s3: Bits received net.if.in["enp0s3"] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Bits sent net.if.out["enp0s3"] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Inbound packets discarded net.if.in["enp0s3",dropped] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Inbound packets with errors net.if.in["enp0s3",errors] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Interface type vfs.file.contents["/sys/class/net/enp0s3/type"] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Operational status vfs.file.contents["/sys/class/net/enp0s3/operstate"] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Outbound packets discarded net.if.out["enp0s3",dropped] Zabbix agent Numeric (unsigned) Enabled
Interface enp0s3: Outbound packets with errors net.if.out["enp0s3",errors] Zabbix agent Numeric (unsigned) Enabled
Select
Cancel
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help

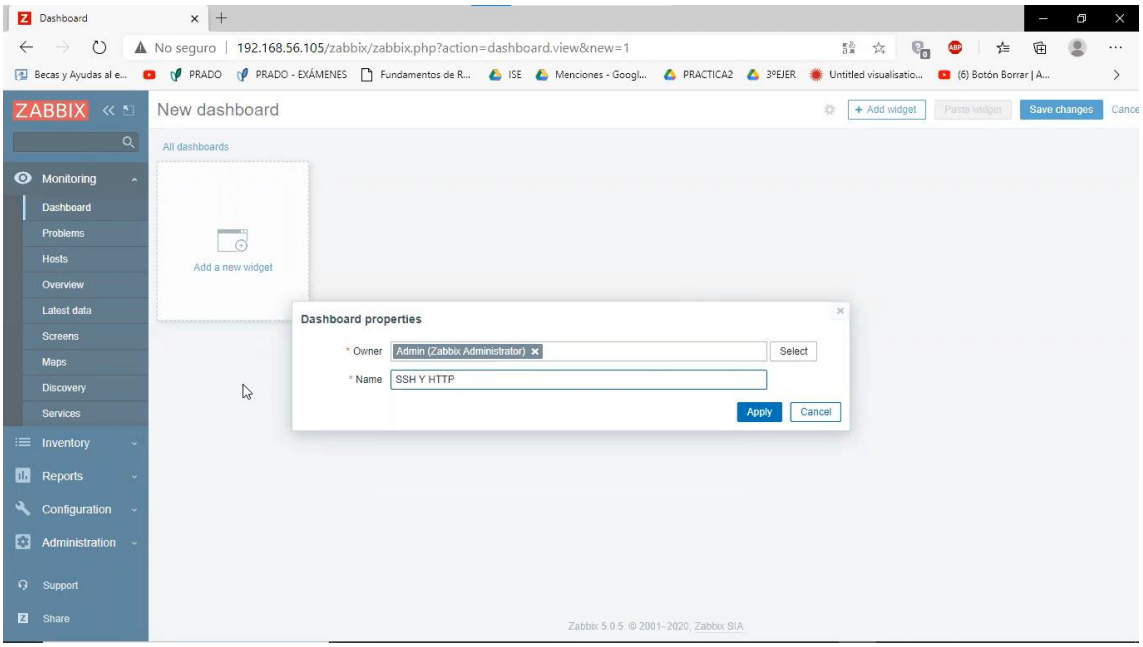

Configuration of graphs
No seguro | 192.168.56.105/zabbix/graphs.php?hostid=10340&form=create
ZABBIX
Graphs
All hosts / UbuntuISE Enabled ZBX SNMP JMX IPMI Applications 20 Items 90 Triggers 36 Graphs 20 Discovery rules 3 Web scenarios
Graph Preview
* Name SSH UBUNTU
* Width 900
* Height 200
Graph type Normal
Show legend
Show working time
Show triggers
Percentile line (left)
Percentile line (right)
Y axis MIN value Calculated
Y axis MAX value Calculated
* Items
Name Function Draw style Y axis side Colour Action
1: UbuntuISE: SSH service is running Remove
Add
Add
Cancel
Monitoring
Inventory
Reports
Configuration
Host groups
Templates
Hosts
Maintenance
Actions
Event correlation
Discovery
Services
Administration
Support
Share
Help

Una vez creadas todas las gráficas, voy a *Monitoring->Dashboard->*(arriba izq.)*All dashboards->create dashboard*

Podemos modificar una existente, pero creo una únicamente para tener las 4 gráficas que quiero.







Añadimos las gráficas anteriormente creadas y podemos ver la monitorización de estas funcionalidades ya.

COMPROBACION

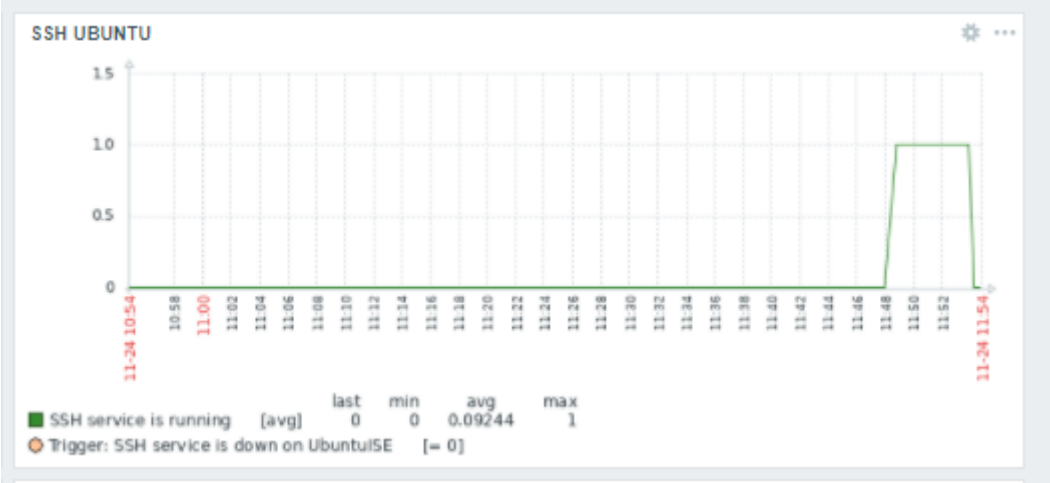
Paro el sistema de sshd y http en centos

```
[javierrp@localhost ~]$ sudo systemctl stop sshd
[sudo] password for javierrp:
[javierrp@localhost ~]$ sudo systemctl stop httpd.service
[javierrp@localhost ~]$
```

Y así se refleja en los gráficos:

Y ahora detengo sshd en Ubuntu y esto es lo que se registra:

```
javierrp@javierrp:~$ sudo systemctl stop sshd
javierrp@javierrp:~$ _
```



ANSIBLE

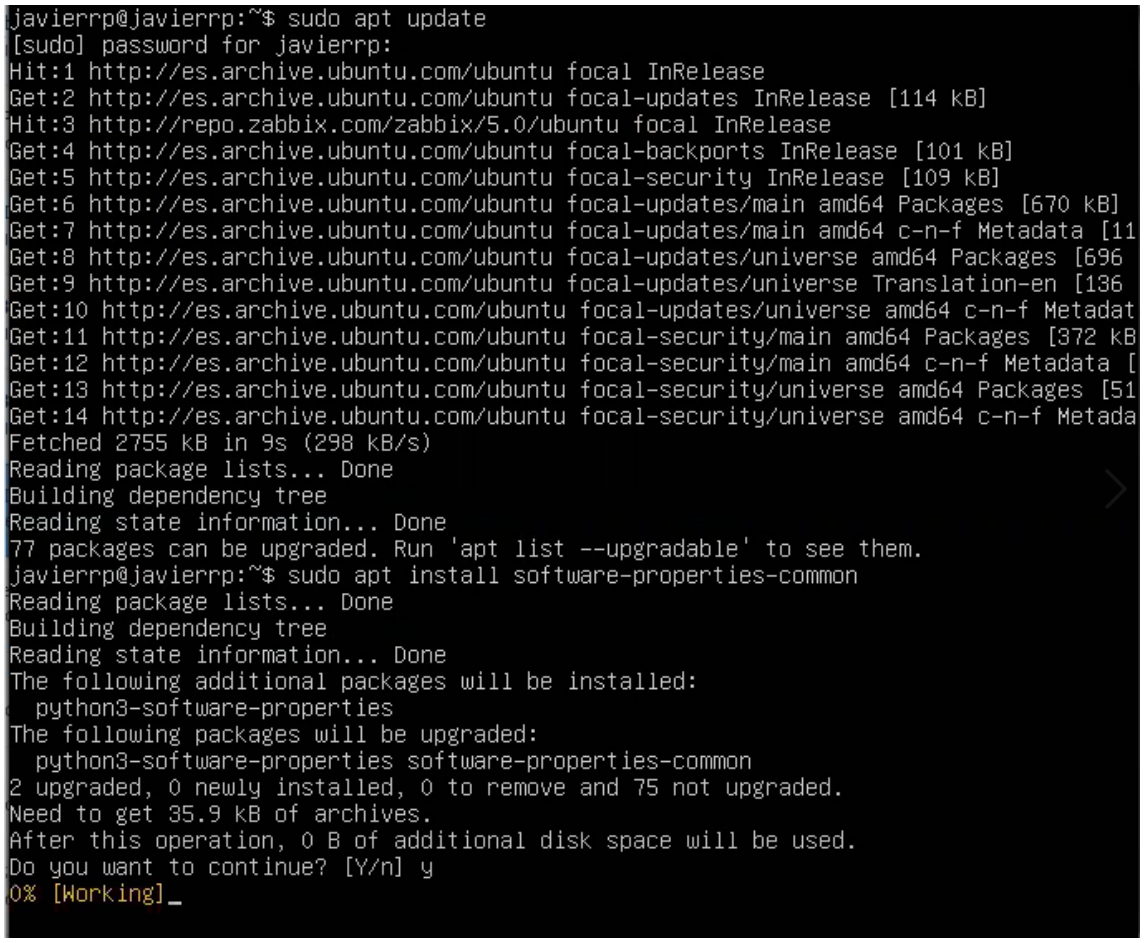
Las fuente usadas para la instalación y prueba de ansible han sido:

- Installing Ansible — Ansible Documentation
- Getting Started — Ansible Documentation

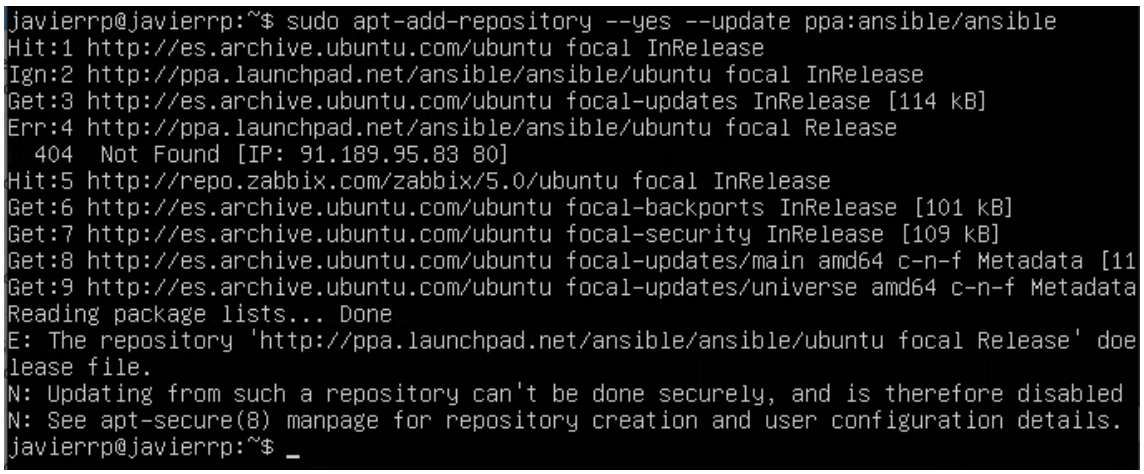
Ansible en Ubuntu Server

Yo utilizaré como servidor mi máquina virtual de Ubuntu Server para asegurarme una correcta configuración de ssh y cortafuegos.

Lo primero es actualizar repositorios:



A estos repositorios añadimos el siguiente:



E instalamos ansible:

```
javierrp@javierrp:~$ sudo apt install ansible
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ieee-data python3-argcomplete python3-crypto python3-dnspython python3-jmespath pyt
  python3-libcloud python3-lockfile python3-netaddr python3-ntlm-auth python3-request
  python3-requests-ntlm python3-selinux python3-winrm python3-xmltodict
Suggested packages:
  cowsay sshpass python-lockfile-doc ipython3 python-netaddr-docs
The following NEW packages will be installed:
  ansible ieee-data python3-argcomplete python3-crypto python3-dnspython python3-jmes
  python3-kerberos python3-libcloud python3-lockfile python3-netaddr python3-ntlm-aut
  python3-requests-kerberos python3-requests-ntlm python3-selinux python3-winrm pytho
0 upgraded, 16 newly installed, 0 to remove and 75 not upgraded.
Need to get 9643 kB of archives.
After this operation, 90.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-crypto amd64 2.6.1
 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-dnspython all 1.16
1 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/main amd64 ieee-data all 20180805.1 [
Get:4 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-netaddr all 0.7.19
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 ansible all 2.9.6+dfsg
30% [5 ansible 921 kB/5794 kB 16%]_
```

Instalación en centos

La instalación en centos es más simple:

```
[javierrp@localhost ~]$ sudo yum install ansible
[sudo] password for javierrp:
Última comprobación de caducidad de metadatos hecha hace 1:28:37, el lun 23 nov 2020
Dependencias resueltas.
================================================================================
 Paquete                    Arquitectura     Versión                 Repositorio
================================================================================
Instalando:
 ansible                    noarch           2.9.15-1.el8            epel
Instalando dependencias:
 libsodium                  x86_64           1.0.18-2.el8            epel
 python3-asn1crypto         noarch           0.24.0-3.el8            BaseOS
 python3-babel              noarch           2.5.1-5.el8             AppStream
 python3-bcrypt             x86_64           3.1.6-2.el8.1           epel
 python3-cffi               x86_64           1.11.5-5.el8            BaseOS
 python3-cryptography       x86_64           2.3-3.el8               BaseOS
 python3-idna               noarch           2.5-5.el8               BaseOS
 python3-jinja2             noarch           2.10.1-2.el8_0          AppStream
 python3-jmespath           noarch           0.9.0-11.el8            AppStream
 python3-markupsafe         x86_64           0.23-19.el8             AppStream
 python3-ply                noarch           3.9-8.el8               BaseOS
 python3-pyasn1             noarch           0.3.7-6.el8             AppStream
 python3-pycparser          noarch           2.14-14.el8             BaseOS
 python3-pynacl             x86_64           1.3.0-5.el8             epel
 python3-pytz               noarch           2017.2-9.el8            AppStream
 sshpass                    x86_64           1.06-9.el8              epel
Instalando dependencias débiles:
 python3-paramiko           noarch           2.4.3-1.el8             epel

Resumen de la transacción
================================================================================
Instalar  18 Paquetes

Tamaño total de la descarga: 25 M
Tamaño instalado: 129 M
¿Está de acuerdo [s/N]?:
```

Configuración de Ubuntu Server para que sea host

Ahora volviendo a Ubuntu Server, que será servidor, modificamos el siguiente archivo:

```
javierrp@javierrp:~$ sudo nano /etc/ansible/hosts
```

Añadiendo al final de este lo siguiente:

```
ubuntuISE ansible_host=192.168.56.105 ansible_user='javierrp'
centosISE ansible_host=192.168.56.110 ansible_user='javierrp'
```

Eso permitirá referirnos a las máquinas con un nombre y no tener que meter sus IP's ni usuarios en cada uso.

Ponemos el puerto en 22022 para tener la misma configuración que el ssh de ambas máquinas:

```
javierrp@javierrp:~$ sudo nano /etc/ansible/ansible.cfg
```

```
#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
```

Y ahora vamos a conectar nuestro host con ambas máquinas remotas. Para este paso, y deforma temporal, deberemos poner el 'permitrootlogin' a 'yes' tanto en centos como en Ubuntu (con su correspondiente "systemctl restart sshd"). Tras tener eso habilitado, creamos una clave pública y privada con ssh-keygen en Ubuntu server y la distribuimos tanto a centos como a ella misma con ssh-copy-id:

```
javierrp@javierrp:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/javierrp/.ssh/id_rsa):
/home/javierrp/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/javierrp/.ssh/id_rsa
Your public key has been saved in /home/javierrp/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:sOobF7Yxg4+Uobfs2d4VWJOTCnR3Lh+C8HeuvYg4Rew javierrp@javierrp
The key's randomart image is:
+---[RSA 3072]----+
|       .+ . o .  |
|         = = =   |
|    . .  .= B +  |
|   . + o oo. B . |
|  . = B So.   o  |
|   + * *  E. o   |
|    * +  .. . .  |
|   o = ..o . . . |
|    =oo o.. . .  |
+----[SHA256]-----+
javierrp@javierrp:~$ ssh-copy-id 192.168.56.105 -p 22022
```

```
javierrp@javierrp:~$ ssh-copy-id 192.168.56.105 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/javierrp/.ssh/id
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out a
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted no
all the new keys
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.

javierrp@javierrp:~$
```

```
javierrp@javierrp:~$ ssh-copy-id 192.168.56.110 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/javierrp/.ssh/id
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out a
eady installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted no
all the new keys
javierrp@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.

javierrp@javierrp:~$ _
```

Tras esto volvemos a poner el 'permitrootlogin no' y reiniciamos el sistema sshd.

Comprobación

Para la comprobación, hacemos un ping a ambas máquinas (no hay que hacerlo por separado, vale con un “ansible all”)

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

Y como funciona, creamos un archivo .py en el directorio /~/ de cada máquina (con el mismo nombre) para probar a ejecutarlo desde el host. Este script se toma del guion de prácticas y sirve para comprobar que los RAID están bien configurados.

Centos:

```
[javierrp@localhost ~]$ sudo nano ~/javi.raid.py_
```

```
import re
f=open('/proc/mdstat')
for line in f:
        b=re.findall('\[[U]*[_]+[U]*\]',line)
        if(b!=[]):
                print("--ERROR_en_RAID--")
print("--OK_Script--")
```

Ubuntu:

```
javierrp@javierrp:~$ sudo nano ~/javi-raid.py _
```

```
  GNU nano 4.8                          /home/javierrp/javi-
import re
f=open('/proc/mdstat')
for line in f:
        b=re.findall('\[[U]*[_]+[U]*\]',line)
        if(b!=[]):
                print("--ERROR_en_RAID--")
print("--OK_Script--")
```

Y ya desde Ubuntu Server como Hosts:

```
javierrp@javierrp:~$ ansible all -a "python3 ~/javi-raid.py"
centosISE | CHANGED | rc=0 >>
--OK_Script--
ubuntuISE | CHANGED | rc=0 >>
--OK_Script--
javierrp@javierrp:~$
```