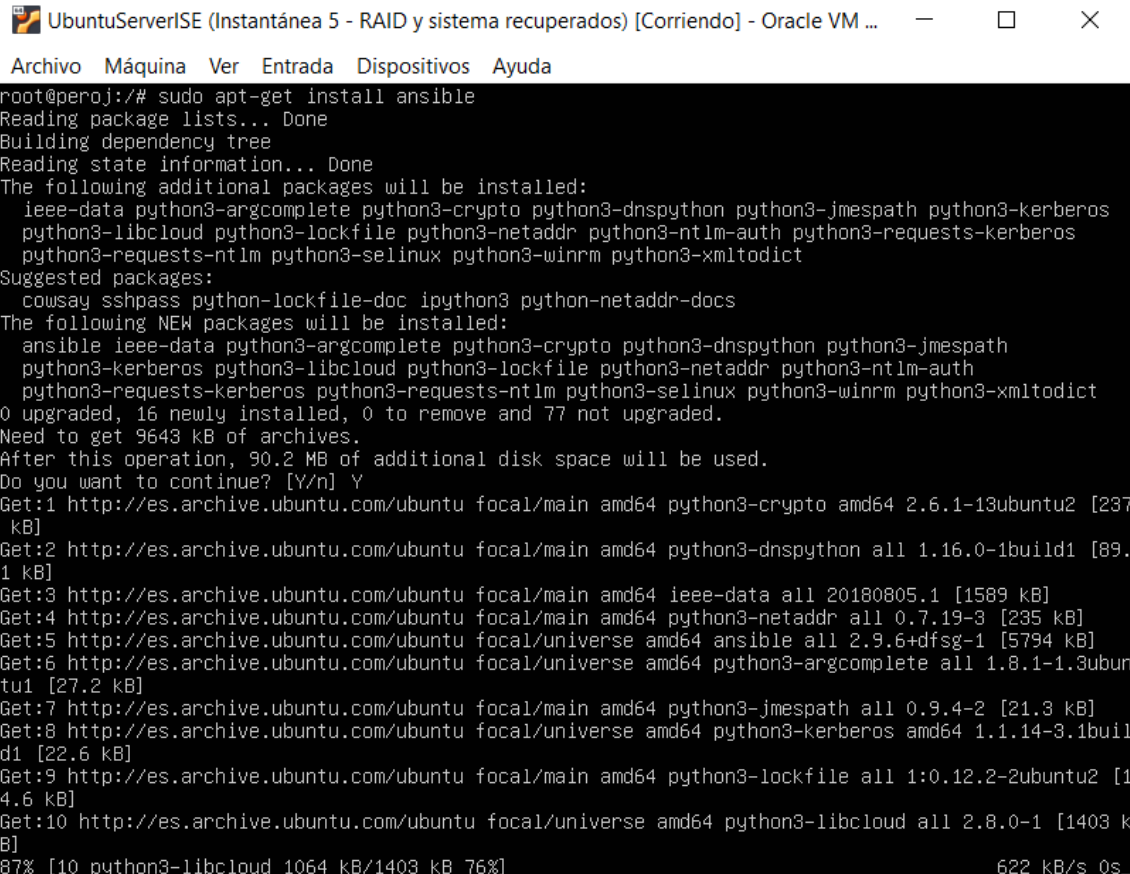


MONITORIZACIÓN Y “PROFILING”

ANSIBLE

Ejercicio 2: Usted deberá saber cómo instalar y configurar Ansible para poder hacer un ping a las máquinas virtuales de los servidores y ejecutar un comando básico (p.ej. el script de monitorización del RAID1). También debe ser consciente de la posibilidad de escribir acciones más complejas mediante playbooks escritos con YAML. Incluya capturas de pantalla del proceso con una breve descripción en el mismo documento que suba para el ejercicio de Zabbix.

1º) Instalamos **ansible** en UbuntuServer con **sudo apt-get install ansible**. Comprobamos la versión con **ansible --version**.

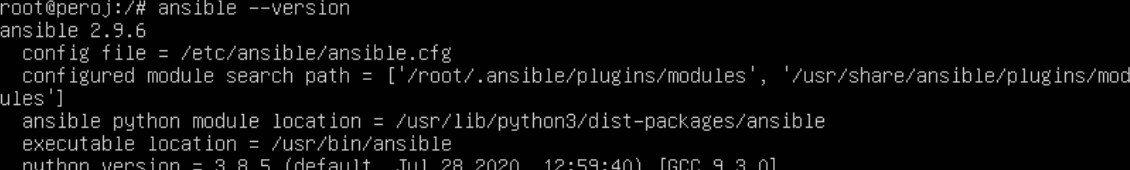


```

root@peroj:~# sudo apt-get 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 python3-kerberos
  python3-libcloud python3-lockfile python3-netaddr python3-ntlm-auth python3-requests-kerberos
  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-jmespath
  python3-kerberos python3-libcloud python3-lockfile python3-netaddr python3-ntlm-auth
  python3-requests-kerberos python3-requests-ntlm python3-selinux python3-winrm python3-xmltodict
0 upgraded, 16 newly installed, 0 to remove and 77 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 2.6.1-13ubuntu2 [237
  kB]
Get:2 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-dnspython all 1.16.0-1build1 [89.
  1 kB]
Get:3 http://es.archive.ubuntu.com/ubuntu focal/main amd64 ieee-data all 20180805.1 [1589 kB]
Get:4 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-netaddr all 0.7.19-3 [235 kB]
Get:5 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 ansible all 2.9.6+dfsg-1 [5794 kB]
Get:6 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-argcomplete all 1.8.1-1.3ubun
  tu1 [27.2 kB]
Get:7 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-jmespath all 0.9.4-2 [21.3 kB]
Get:8 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-kerberos amd64 1.1.14-3.1buil
  d1 [22.6 kB]
Get:9 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python3-lockfile all 1:0.12.2-2ubuntu2 [1
  4.6 kB]
Get:10 http://es.archive.ubuntu.com/ubuntu focal/universe amd64 python3-libcloud all 2.8.0-1 [1403 k
  B]
87% [10 python3-libcloud 1064 kB/1403 kB 76%]
622 kB/s 0s_

```

Y vemos que usa la versión 3.8.5 de Python, es decir, el módulo **python3** para ejecutar los scripts **.py**.



```

root@peroj:~# ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/mod
  ules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jul 28 2020, 12:59:40) [GCC 9.3.0]

```

2º) Ahora deberemos irnos al archivo de inventario con **vi /etc/ansible/hosts** donde indicaremos los host con las IPs de los servidores con los que queremos tratar, en nuestro caso,

tanto de Ubuntu como de CentOS (podemos ver varios ejemplos de configuración comentados en el archivo).

Este archivo se suele utilizar también para configurar variables que serán válidas sólo para hosts o grupos específicos, a fin de usarse dentro de los *playbooks* y las plantillas. Algunas variables también pueden afectar la forma en que se ejecuta un *playbook*, como la variable *ansible_python_interpreter* que veremos a continuación.

```
root@peroj:/# ls /etc/ansible/
ansible.cfg  hosts
root@peroj:/# vi /etc/ansible/hosts_
```

Creamos un grupo de servidores llamado *auto_raids* donde incluiremos las IPs tanto de CentOS como de Ubuntu. También creamos (si es necesario) un subgrupo de este llamado *auto_raids:vars* donde establecemos el parámetro *ansible_python_interpreter=/usr/bin/python3* (descomentar sólo si es necesario).

```
# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups

#Grupo para servidores Ubuntu y CentOS
[auto_raids]
ubuntu_server ansible_host=192.168.56.105
centos_server ansible_host=192.168.56.110

#Subgrupo que establece el parametro que garantiza que el servidor remoto utilice python3
#[auto_raids:vars]
#ansible_python_interpreter=/usr/bin/python3
```

Comprobamos que se han añadido los host correctamente al inventario con ***ansible-inventory -list -y***.

```
root@peroj:/# ansible-inventory --list -y
all:
  children:
    auto_raids:
      hosts:
        centos_server:
          ansible_host: 192.168.56.110
        ubuntu_server:
          ansible_host: 192.168.56.105
      ungrouped: {}
```

3º) Antes de probar conexión, debemos cambiar el puerto de las máquinas al puerto por defecto (22), con lo que esto conlleva, en Ubuntu deberemos simplemente avisar al firewall, pero en CentOS también deberemos actualizar el SELinux y ya después, volver a generar los pares de claves pública-privada, tanto de Ubuntu consigo mismo como con CentOS (ya que *ansible* se conecta por ssh).

-En Ubuntu: Abrimos `/etc/ssh/sshd_config` y cambiamos el puerto al *Port 22* y el parámetro *Password Authentication* *yes* para poder generar las nuevas claves (después lo cambiamos otra vez). Después de guardar el fichero, reiniciamos el servicio con **`systemctl restart sshd.service`** para aplicar los cambios y lo vemos con **`systemctl status sshd.service`**. Nos aseguramos que el puerto 22 está abierto en el firewall **`ufw allow 22`** (que modifica las *iptables* para que nos permita comunicación por este puerto).

```
peroj@peroj:/$ systemctl restart sshd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'sshd.service'.
Authenticating as: peroj
Password:
==== AUTHENTICATION COMPLETE ====
peroj@peroj:/$ systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-11-18 20:44:18 UTC; 12s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 5666 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 5681 (sshd)
     Tasks: 1 (limit: 4621)
    Memory: 1.1M
   CGroup: /system.slice/sshd.service
           └─5681 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Nov 18 20:44:18 peroj systemd[1]: Starting OpenBSD Secure Shell server...
Nov 18 20:44:18 peroj sshd[5681]: Server listening on 0.0.0.0 port 22.
Nov 18 20:44:18 peroj sshd[5681]: Server listening on :: port 22.
Nov 18 20:44:18 peroj systemd[1]: Started OpenBSD Secure Shell server.
peroj@peroj:/$ ssh-keygen
```

Primero generamos la clave en el cliente (Ubuntu) con **`ssh-keygen`**. Se la mandamos al servidor (Ubuntu) con **`ssh-copy-id peroj@192.168.56.105 -p 22`**, metemos la contraseña de Ubuntu y vemos que se ha añadido una clave. Finalmente, comprobamos haciendo login por **`ssh peroj@192.168.56.105 -p 22`**, ya no nos pide la contraseña.

```
peroj@peroj:/$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/peroj/.ssh/id_rsa):
/home/peroj/.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/peroj/.ssh/id_rsa
Your public key has been saved in /home/peroj/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Q+aRlt7bHbc2p0Za9cdAx2IurMCiCItKJwzVRypeWBB peroj@peroj
The key's randomart image is:
+---[RSA 3072]-----+
|    .E      .    |
|  .O.O..  o    + o |
| .O O...B   . + o |
|o. O  .*oo  o o . |
|+o..  .S... ..+o |
|o+...  ..o .oo= |
|o o      . .+.+o |
| .        . o.o |
| ..       ..    |
+---[SHA256]-----+
peroj@peroj:/$ ssh-copy-id peroj@192.168.56.105 -p 22
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/peroj/.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
peroj@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '22' 'peroj@192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.
peroj@peroj:/$
```

```

peroj@peroj:/$ sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
peroj@peroj:/$ ssh peroj@192.168.56.105 -p 22
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Nov 18 20:50:43 UTC 2020

System load:  0.0               Processes:            141
Usage of /home: 0.2% of 975MB   Users logged in:     1
Memory usage:  15%             IPv4 address for enp0s3: 10.0.2.15
Swap usage:    0%              IPv4 address for enp0s8: 192.168.56.105

 * Introducing self-healing high availability clustering for MicroK8s!
   Super simple, hardened and opinionated Kubernetes for production.

   https://microk8s.io/high-availability

80 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

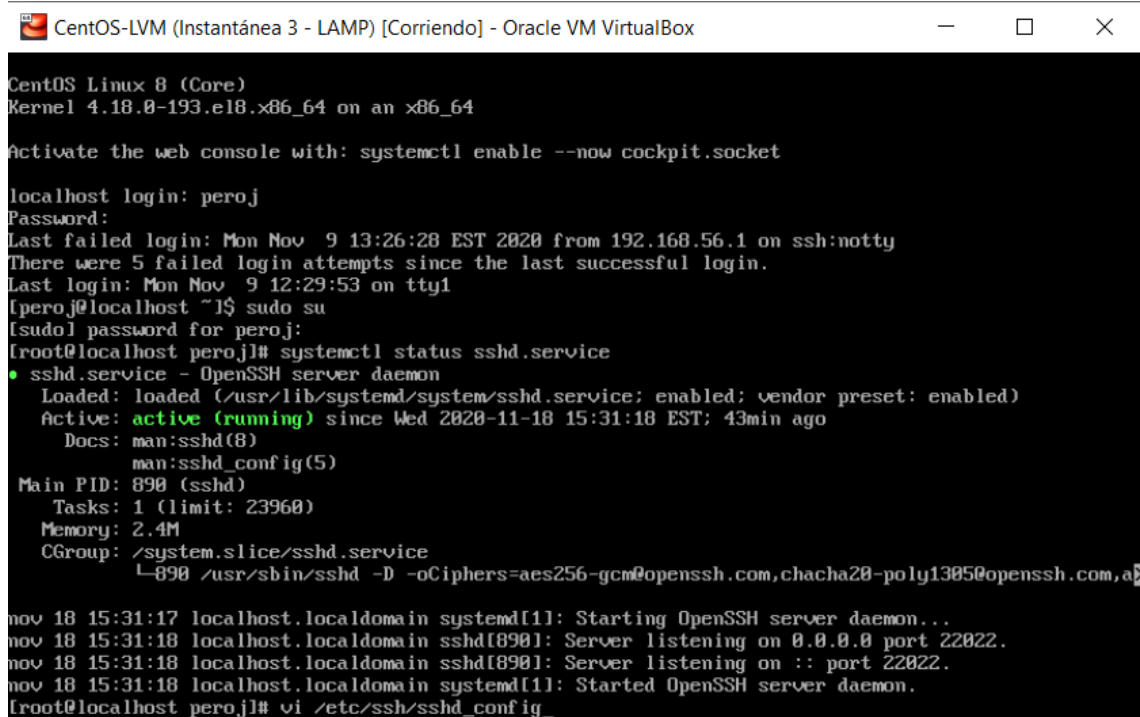
Last login: Wed Nov 18 16:53:30 2020
peroj@peroj:~$ exit
logout
Connection to 192.168.56.105 closed.
peroj@peroj:/$

```

Ya tenemos conexión por ssh de Ubuntu consigo misma a través del puerto 22.

En Ubuntu debemos recordar poner el parámetro de */etc/ssh/sshd_config* *AuthenticationPassword no*.

-En CentOS: Haremos “el mismo” proceso anterior para configurarle el ssh server y podremos conectar desde UbuntuServer. Comprobamos que esta dado de alta el servicio con **`sudo systemctl status sshd.service`**. Vamos a cambiarle también al puerto 22 y el **`PasswordAuthentication`** a **`yes`** modificando el fichero **`sudo vi /etc/ssh/sshd_config`**.



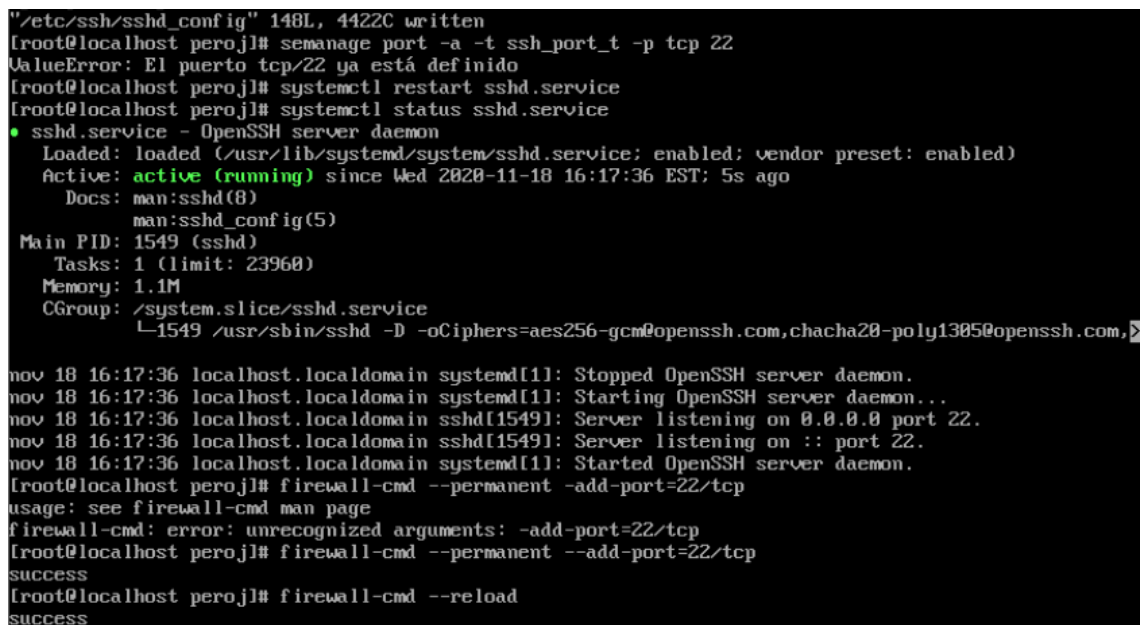
```
CentOS-LVM (Instantánea 3 - LAMP) [Corriendo] - Oracle VM VirtualBox
CentOS Linux 8 (Core)
Kernel 4.18.0-193.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

localhost login: peroj
Password:
Last failed login: Mon Nov  9 13:26:28 EST 2020 from 192.168.56.1 on ssh:notty
There were 5 failed login attempts since the last successful login.
Last login: Mon Nov  9 12:29:53 on tty1
[peroj@localhost ~]$ sudo su
[sudo] password for peroj:
[root@localhost peroj]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-11-18 15:31:18 EST; 43min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 890 (sshd)
    Tasks: 1 (limit: 23960)
   Memory: 2.4M
   CGroup: /system.slice/sshd.service
           └─890 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,a
nov 18 15:31:17 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
nov 18 15:31:18 localhost.localdomain sshd[890]: Server listening on 0.0.0.0 port 22022.
nov 18 15:31:18 localhost.localdomain sshd[890]: Server listening on :: port 22022.
nov 18 15:31:18 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost peroj]# vi /etc/ssh/sshd_config_
```

Una vez hecho esto, debemos avisar al sistema de seguridad de Linux SELinux de que hemos cambiado el puerto con la orden **`semanage port -a -t ssh_port_t -p tcp 22`** y reiniciamos el servicio con **`sudo systemctl restart sshd.service`**.

También avisamos al cortafuegos de que permita conexiones por ese puerto con **`sudo firewall-cmd --permanent --add-port=22022/tcp`** y recargamos las reglas con **`sudo firewall-cmd --reload`**.



```
"/etc/ssh/sshd_config" 148L, 4422C written
[root@localhost peroj]# semanage port -a -t ssh_port_t -p tcp 22
ValueError: El puerto tcp/22 ya está definido
[root@localhost peroj]# systemctl restart sshd.service
[root@localhost peroj]# systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-11-18 16:17:36 EST; 5s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Main PID: 1549 (sshd)
    Tasks: 1 (limit: 23960)
   Memory: 1.1M
   CGroup: /system.slice/sshd.service
           └─1549 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,a
nov 18 16:17:36 localhost.localdomain systemd[1]: Stopped OpenSSH server daemon.
nov 18 16:17:36 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
nov 18 16:17:36 localhost.localdomain sshd[1549]: Server listening on 0.0.0.0 port 22.
nov 18 16:17:36 localhost.localdomain sshd[1549]: Server listening on :: port 22.
nov 18 16:17:36 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost peroj]# firewall-cmd --permanent --add-port=22/tcp
usage: see firewall-cmd man page
firewall-cmd: error: unrecognized arguments: --add-port=22/tcp
[root@localhost peroj]# firewall-cmd --permanent --add-port=22/tcp
success
[root@localhost peroj]# firewall-cmd --reload
success
```

Ya si intentamos conectarnos desde Ubuntu a CentOS con **`ssh peroj@192.168.56.110 -p 22`** ya podemos sin problema.

Ahora debemos enviar desde Ubuntu la clave pública-privada a CentOS para que no sea necesaria la contraseña.

```
peroj@peroj:/$ ssh-copy-id peroj@192.168.56.110 -p 22
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/peroj/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
peroj@192.168.56.110's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '22' 'peroj@192.168.56.110'"
and check to make sure that only the key(s) you wanted were added.

peroj@peroj:/$ ssh peroj@192.168.56.110 -p 22
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Nov 18 16:30:59 2020 from 192.168.56.105
[peroj@localhost ~]$ exit
logout
Connection to 192.168.56.110 closed.
peroj@peroj:/$
```

Y finalmente, le cambiamos del fichero de configuración de CentOS ***sudo vi /etc/ssh/sshd_config*** el *PasswordAuthentication* a no, para que solo se pueda conectar un usuario que previamente esté vinculado con una clave pública-privada. Y reiniciamos el servicio con ***sudo systemctl restart sshd.service***.

```
"/etc/ssh/sshd_config" 148L, 4422C written
[root@localhost peroj]# systemctl restart sshd.service
[root@localhost peroj]# _
```

UbuntuServerISE (Instantánea 5 - RAID y sistema recuperados) [Corriendo] - C

Archivo Máquina Ver Entrada Dispositivos Ayuda

```
peroj@peroj:/$ ssh peroj@192.168.56.110 -p 22
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Nov 18 16:31:51 2020 from 192.168.56.105
[peroj@localhost ~]$ exit
logout
Connection to 192.168.56.110 closed.
```

```

UbuntuServerISE (Instantánea 5 - RAID y sistema recuperados) [Corriendo] - Oracle
Archivo Máquina Ver Entrada Dispositivos Ayuda
peroj@peroj:/$ ssh peroj@192.168.56.105 -p 22
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-52-generic x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Wed Nov 18 21:39:02 UTC 2020

System load: 0.0          Processes: 140
Usage of /home: 0.2% of 975MB   Users logged in: 1
Memory usage: 15%          IPv4 address for enp0s3: 10.0.2.15
Swap usage: 0%             IPv4 address for enp0s8: 192.168.56.105

* Introducing self-healing high availability clustering for MicroK8s!
  Super simple, hardened and opinionated Kubernetes for production.

  https://microk8s.io/high-availability

80 updates can be installed immediately.
0 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Wed Nov 18 21:35:15 2020 from 192.168.56.105
peroj@peroj:~$ exit
logout
Connection to 192.168.56.105 closed.
peroj@peroj:/$ ssh peroj@192.168.56.110 -p 22
Activate the web console with: systemctl enable --now cockpit.socket

Last login: Wed Nov 18 16:35:08 2020 from 192.168.56.105
[peroj@localhost ~]$ exit
logout
Connection to 192.168.56.110 closed.

```

4º) Una vez que tenemos correctamente configuradas las conexiones ssh procedemos a comprobar que Ansible nos devuelve el ping correctamente a los servidores con ***ansible all -m ping -u peroj*** o ***ansible auto_raids -m ping -u peroj***.

```

UbuntuServerISE (Instantánea 5 - RAID y sistema recuperados) [Corriendo] - Oracle
Archivo Máquina Ver Entrada Dispositivos Ayuda
peroj@peroj:/$ ansible all -m ping -u peroj
ubuntu_server | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
centos_server | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
peroj@peroj:/$ ansible auto_raids -m ping -u peroj
ubuntu_server | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
centos_server | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}

```

5º) Ahora debemos crear el script **mon_raid.py** en CentOS (y en la misma ubicación que en Ubuntu). También podemos enviarlo por ssh desde Ubuntu a CentOS.

Ahora podemos ejecutarlo con Ansible en todos los servidores y recibir el resultado en nuestra máquina con **ansible all -a "python3 /home/perorj/mon_raid.py"** (si el fichero a ejecutar no estuviera en la misma ubicación en las dos máquinas, esto deberíamos hacerlo con un script de configuración o *playbook*). Comprobamos y vemos como en Ubuntu nos devuelve el error en los RAID pero en CentOS no, dado que en CentOS (aunque es la versión sin RAID) funcionan correctamente los discos duros.



The image shows two terminal windows. The top window is titled 'CentOS-LVM (Instantánea 3 - LAMP) [Corriendo] - Oracle VM VirtualBox' and shows a root user at localhost perojl# running 'ls mon_raid.py' which returns 'UbuntuBackup'. The bottom window is titled 'UbuntuServerISE (Instantánea 5 - RAID y sistema recuperados) [Corrienc' and shows a user at peroj:~\$ running 'ansible all -a "python3 /home/perorj/mon_raid.py"'. The output shows 'centos_server' with 'CHANGED' and 'rc=0', followed by a series of empty lists and 'END_OF_SCRIPT'. For 'ubuntu_server', it shows 'CHANGED' and 'rc=0', followed by a list containing '[U_]', an 'ERROR_EN_RAID--' message, another list with '[U_]', another 'ERROR_EN_RAID--' message, and finally 'END_OF_SCRIPT'.

```
CentOS-LVM (Instantánea 3 - LAMP) [Corriendo] - Oracle VM VirtualBox
root@localhost perojl# ls
mon_raid.py  UbuntuBackup
root@localhost perojl# _

UbuntuServerISE (Instantánea 5 - RAID y sistema recuperados) [Corrienc
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
peroj@peroj:~$ ansible all -a "python3 /home/perorj/mon_raid.py"
centos_server | CHANGED | rc=0 >>
[]
[]
--END_OF_SCRIPT
ubuntu_server | CHANGED | rc=0 >>
[]
[]
['[U_]']
--ERROR_EN_RAID--
[]
[]
['[U_]']
--ERROR_EN_RAID--
[]
[]
--END_OF_SCRIPT
peroj@peroj:~$
```


ZABBIX

Ejercicio 1:

Realice una instalación de Zabbix 5.0 en su servidor con Ubuntu Server 20.04 y configure para que se monitorice a él mismo y para que monitorice a la máquina con CentOS. Puede configurar varios parámetros para monitorizar, uso de CPU, memoria, etc. pero debe configurar de manera obligatoria la monitorización de los servicios SSH y HTTP. Documente el proceso de instalación y configuración indicando las referencias que ha utilizado así como los problemas que ha encontrado. Para ello puede usar cualquier tipo de formato de documento (respetando claridad y corrección) y procure que en las capturas aparezca su nombre de usuario (en el prompt p.ej.). El archivo debe estar subido a SWAD (zona mis trabajos) antes del examen de esta práctica.

(toda la información obtenida durante la realización del ejercicio ha sido del manual oficial de Zabbix 5.0)

1. Conceptos básicos Zabbix

Es un software de monitorización que crea un *frontend* en el navegador que permite ver de forma visual los envíos de los agentes (la información que envían los servidores monitorizados).

Tenemos lo que se conoce como:

- Servidor --> Es el que tiene la recolección de datos. Procesa los datos para monitorizar lo que deseamos de los agentes. Vamos a tener uno, Ubuntu.
- Agentes --> Son los que le envían los datos al servidor. Vamos a tener dos agentes, CentOS y Ubuntu.

Una vez que instalemos y configuremos Zabbix tanto en CentOS como en Ubuntu, trabajaremos siempre desde el *frontend* en el servidor Ubuntu.

ARQUITECTURA ZABBIX

Zabbix está formado por varios componentes software:

- SERVER --> Zabbix Server es el componente central al que los agentes reportan o devuelven la información y estadísticas de disponibilidad e integridad. El servidor es el repositorio central en el que se almacenan todos los datos de configuración, estadísticos y operativos.
- DB STORAGE --> Toda la información de configuración, así como los datos recopilados por Zabbix, se almacenan en una base de datos.
- WEB INTERFACE --> Para un fácil acceso a Zabbix desde cualquier lugar y desde cualquier plataforma, se proporciona la interfaz web. La interfaz es parte del servidor Zabbix y generalmente (pero no necesariamente) se ejecuta en la misma máquina física que la que ejecuta el servidor.
- PROXY --> El proxy Zabbix puede recopilar datos de rendimiento y disponibilidad en nombre del servidor Zabbix. Un proxy es una parte opcional de la implementación de Zabbix; sin embargo, puede ser muy beneficioso distribuir la carga de un solo servidor Zabbix. (No vamos a usarlos aquí)

- AGENT --> Los agentes Zabbix se implementan en objetivos de monitoreo para monitorear activamente los recursos y aplicaciones locales e informar de los datos recopilados al servidor Zabbix. Desde Zabbix4.4, hay dos tipos de agentes disponibles:
 - o Zabbix Agent: Ligero, compatible con muchas plataformas, escrito en C.
 - o Zabbix Agent2: Extraflexible, fácilmente ampliable con complementos, escrito en Go.

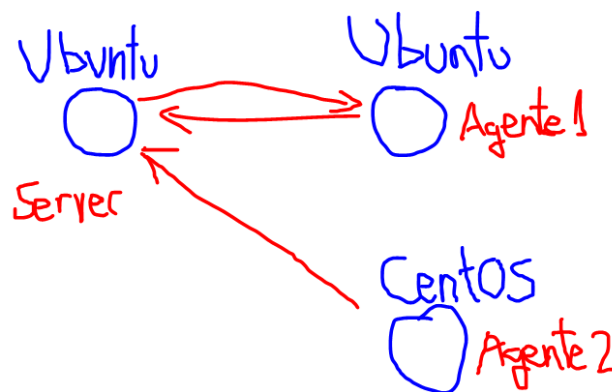
FLUJO DE DATOS GENERAL

Para crear un elemento que recopile datos, primero debe crear un *host*. Pasando al otro extremo del espectro de Zabbix, primero debemos tener un *item* (elemento) para crear un disparador. Debemos tener un disparador para crear una acción (*action*). Por lo tanto, si deseamos recibir, por ejemplo, una alerta de que la CPU está sobrecargada en el servidorX, primero debemos crear una entrada de host (*host entry*) para el servidorX seguida de un *item* para monitorear la CPU, luego un disparador (*trigger*) que se activa si la CPU es demasiado alta, seguido de un *action* que le envía un correo electrónico, por ejemplo.

Aunque parezcan muchos pasos, con el uso de plantillas realmente no son tantos. Sin embargo, gracias a este diseño es posible crear configuraciones muy flexibles.

INSTALACIÓN

<https://www.zabbix.com/download>



Por tanto, deberemos instalar tanto el servidor como el agente en la máquina con UbuntuServer. Y sólo el agente en CentOS.

2. Instalación y Configuración en Ubuntu - Zabbix Server y Zabbix Agent:

1 Elige tu plataforma

VERSIÓN ZABBIX	DISTRIBUCIÓN DE SO	VERSIÓN DEL SISTEMA OPERATIVO	BASE DE DATOS	SERVIDOR WEB
5.2	Red Hat Enterprise Linux	20.04 (Focal)	MySQL	Apache
5.0 LTS	CentOS	18.04 (Bionic)	PostgreSQL	NGINX
4.0 LTS	Oracle Linux	16.04 (Xenial)		
3.0 LTS	Ubuntu	14.04 (Trusty)		
	Debian			
	SUSE Linux Enterprise Server			
	Raspbian			

Y seguimos los pasos que se nos indica más abajo.

a) Instalar el repositorio de Zabbix:

```
Last login: Wed Nov 18 21:52:29 UTC 2020 from 192.168.56.105 on pts/2
peroj@peroj:~$ sudo su
[sudo] password for peroj:
root@peroj:/home/peroj# wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
--2020-11-21 10:44:40-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_5.0-1+focal_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 162.243.159.138, 2604:a880:1:20::b82:1001
Connecting to repo.zabbix.com (repo.zabbix.com)|162.243.159.138|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4244 (4.1K) [application/octet-stream]
Saving to: 'zabbix-release_5.0-1+focal_all.deb'

zabbix-release_5.0-1+foc 100%[=====] 4.14K --.-KB/s in 0s
2020-11-21 10:44:40 (215 MB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]

root@peroj:/home/peroj# dpkg -i zabbix-release_5.0-1+focal_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 118323 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) ...
root@peroj:/home/peroj# apt update_
```

b) Instalar el servidor, la interfaz y el agente de Zabbix:

UbuntuServerISE (Instantánea 6 - Configurado Ansible y reconfigurado ssh) [Corriendo] ...

```
root@peroj:/home/peroj# apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent
```

- c) Crear base de datos inicial en el host de base de datos:

```

root@peroj:/home/peroj# mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.22-0ubuntu0.20.04.2 (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.

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 'practicas,ISE';
Query OK, 0 rows affected (0.02 sec)

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

mysql> quit;
Bye

```

En el servidor Zabbix, importamos el esquema inicial y los datos iniciales. Se pide que ingresemos contraseña recién creada (*practicas,ISE*).

```

root@peroj:/home/peroj# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p z
abbix
Enter password:
root@peroj:/home/peroj# _

```

- d) Configurar la base de datos para el servidor Zabbix. Editamos el archivo **/etc/zabbix/zabbix_server.conf**:

```

DBUser=zabbix

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

```

- e) Configurar PHP para la interfaz de Zabbix. Editamos el archivo **/etc/zabbix/apache.conf**, descomentamos y configuramos nuestra zona horaria:

```

<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>

```

- f) Iniciamos los procesos del agente y del servidor Zabbix y los configuramos para que se inicien con el sistema:

```
root@peroj:/home/peroj# systemctl restart zabbix-server zabbix-agent apache2
root@peroj:/home/peroj# 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.
root@peroj:/home/peroj# systemctl status zabbix-server zabbix-agent apache2
```

```
• zabbix-server.service - Zabbix Server
  Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-11-20 21:02:02 UTC; 1min 8s ago
  Main PID: 6356 (zabbix_server)
  Tasks: 1 (limit: 4621)
  Memory: 3.7M
  CGroup: /system.slice/zabbix-server.service
          └─6356 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf

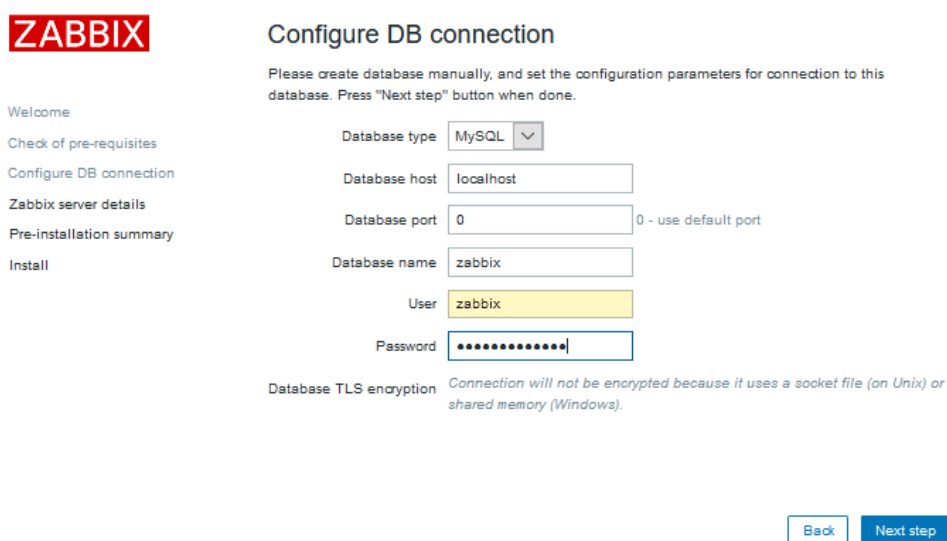
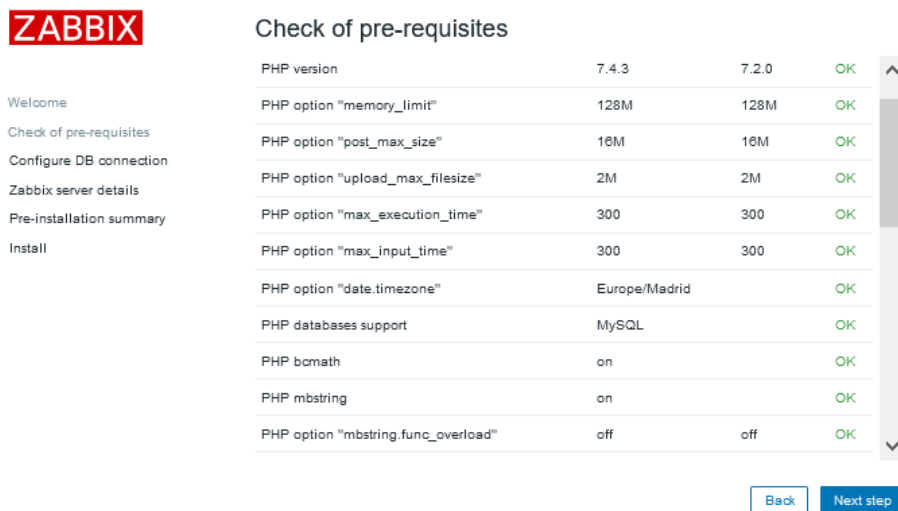
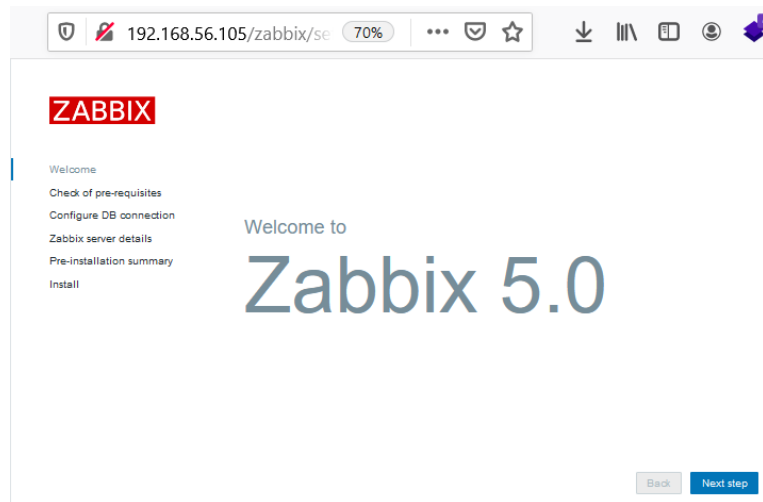
Nov 20 21:02:02 peroj systemd[1]: Starting Zabbix Server...
Nov 20 21:02:02 peroj systemd[1]: Started Zabbix Server.

• zabbix-agent.service - Zabbix Agent
  Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-11-20 21:02:02 UTC; 1min 8s ago
  Main PID: 6358 (zabbix_agentd)
  Tasks: 6 (limit: 4621)
  Memory: 4.4M
  CGroup: /system.slice/zabbix-agent.service
          └─6358 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
              └─6359 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
                  └─6360 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                      └─6361 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                          └─6362 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                              └─6363 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

Nov 20 21:02:02 peroj systemd[1]: Starting Zabbix Agent...
Nov 20 21:02:02 peroj systemd[1]: Started Zabbix Agent.

• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2020-11-20 21:02:03 UTC; 1min 8s ago
  Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 6370 (apache2)
  Tasks: 6 (limit: 4621)
  Memory: 12.0M
```

- g) Configurar la interfaz de Zabbix. Nos conectamos a la interfaz Zabbix recién instalada a <http://192.168.56.105/zabbix>:



Password: *practicas,ISE*



Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

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

Port

Name



Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Pre-installation summary

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

Database type **MySQL**

Database server **localhost**

Database port **default**

Database name **zabbix**

Database user **zabbix**

Database password *********

Database TLS encryption **false**

Zabbix server **localhost**

Zabbix server port **10051**

Zabbix server name **Zabbix-ISE**

[Back](#) [Next step](#)



Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

Pre-installation summary

Install

Install

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "/usr/share/zabbix/conf/zabbix.conf.php" created.

[Back](#) [Finish](#)

ZABBIX

Username

Admin

Login name or password is incorrect.

Password

•••••

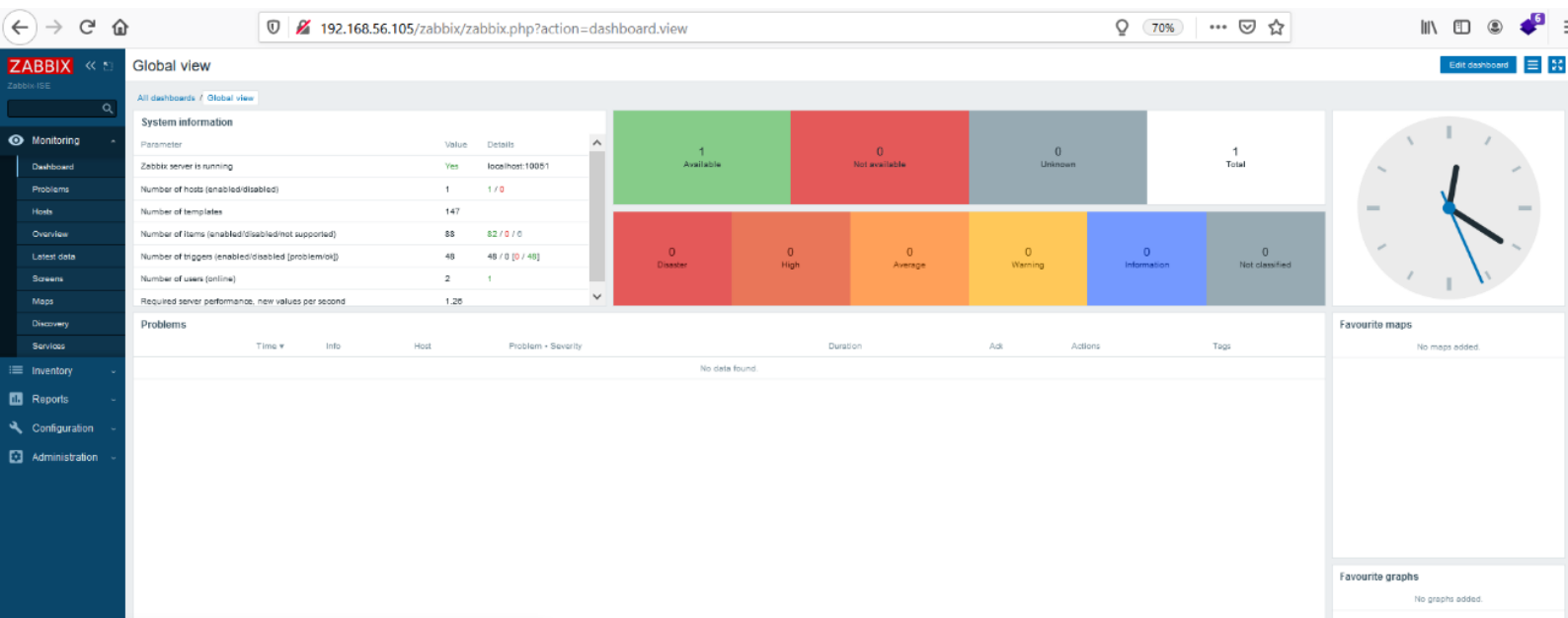
☒ Remember me for 30 days

Sign in

Username: Admin

Password: zabbix

Si introducimos 5 veces mal los datos de acceso la interfaz de zabbix se pausará durante 30 minutos para prevenir ataques de fuerza bruta. Las IPs de los intentos fallidos se mostrarán al iniciar sesión correctamente.



Probamos a añadir un usuario nuevo.

Fuente: <https://www.zabbix.com/documentation/5.0/manual/quickstart/login>

ZABBIX << Zabbix-ISE

Users

User Media Permissions

* Alias: DebugUser

Name: Pepito

Surname: Debugger

* Groups: Enabled debug mode X type here to search Select

* Password: *****

* Password (once again): *****

Password is not mandatory for non internal authentication type.

Language: English (en_GB) You are not able to choose some of the languages, because locales for them are not installed on the web server.

Theme: High-contrast dark

Auto-login: ☐

Auto-logout: ☐ 15m

* Refresh: 30s

* Rows per page: 50

URL (after login):

Add Cancel

Contraseña para el nuevo usuario *pepito*: *debugger*

Users

User group: All Create user

User added

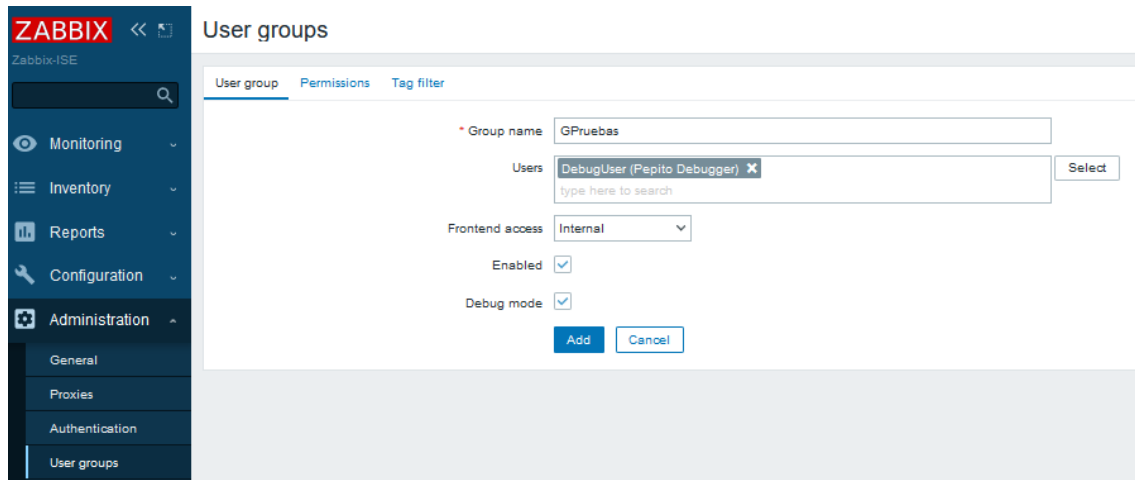
Filter

Alias	Name	Surname	User type	Groups	Is online?	Login	Frontend access	Debug mode	Status
<input type="checkbox"/> Admin	Zabbix	Administrator	Zabbix Super Admin	Zabbix administrators	Yes (2020-11-21 13:10:34)	Ok	System default	Disabled	Enabled
<input type="checkbox"/> DebugUser	Pepito	Debugger	Zabbix Admin	Enabled debug mode	No	Ok	System default	Enabled	Enabled
<input type="checkbox"/> guest			Zabbix User	Disabled, Guests	No	Ok	Internal	Disabled	Disabled

Displaying 3 of 3 found

Por defecto, los usuarios nuevos no tienen permisos de acceso al host.

Entonces, debemos saber que los permisos solo se pueden asignar a grupos de usuarios no a usuarios individuales. Vamos a probar a crear un nuevo grupo de usuarios donde esté el usuario *pepito* y que adquiriera acceso de solo lectura al grupo de servidores de Linux.



ZABBIX << Zabbix-ISE

User groups

User group Permissions Tag filter

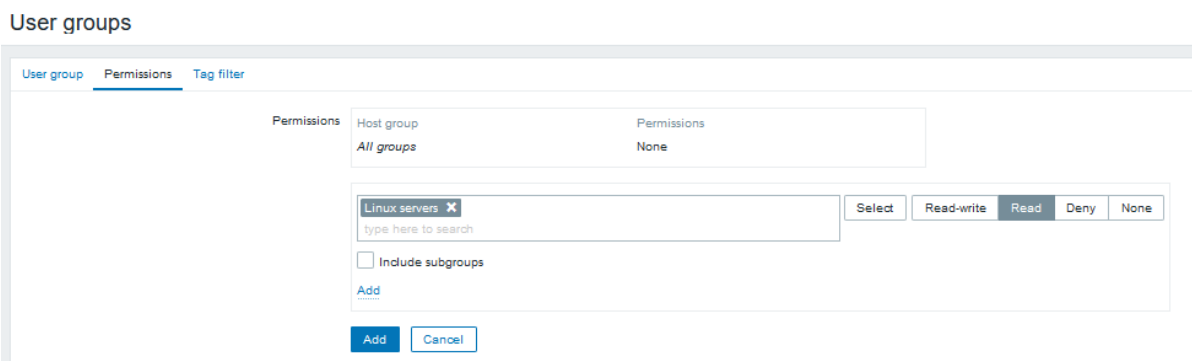
* Group name:

Users:

Frontend access:

Enabled: ☒

Debug mode: ☒



User groups

User group Permissions Tag filter

Permissions

Host group:

Permissions:

☐ Include subgroups

User groups

✓ User group added

Name: Status:

<input type="checkbox"/>	Name	#	Members	Frontend access	Debug mode	Status
<input type="checkbox"/>	Disabled	Users: 1	guest	System default	Disabled	Disabled
<input type="checkbox"/>	Enabled debug mode	Users: 1	DebugUser (Pepito Debugger)	System default	Enabled	Enabled
<input type="checkbox"/>	GPuebas	Users: 1	DebugUser (Pepito Debugger)	Internal	Enabled	Enabled
<input type="checkbox"/>	Guests	Users: 1	guest	Internal	Disabled	Enabled
<input type="checkbox"/>	No access to the frontend	Users: 1		Disabled	Disabled	Enabled
<input type="checkbox"/>	Zabbix administrators	Users: 1	Admin (Zabbix Administrator)	System default	Disabled	Enabled

0 selected

Displaying 6 of 6 found

Para añadir nuevas máquinas a monitorear nos vamos a *Configuracion/Hosts/CreateHost* como se indica en el manual de Zabbix5.0: <https://www.zabbix.com/documentation/5.0/manual/quickstart/host>

Un host en Zabbix es una entidad en red (física, virtual) que desea monitorizar. La definición de lo que puede ser un “host” en Zabbix es bastante flexible ya que puede ser un servidor físico, un conmutador de red, una máquina virtual o alguna aplicación.

Por defecto vemos que ya tenemos creado el host *Zabbix server*. Pero debemos modificarle la dirección IP para indicar que el servidor Zabbix está en nuestra máquina Ubuntu Server.

Vamos a añadir un nuevo host, nuestra máquina virtual Ubuntu (agente).

The screenshot shows the Zabbix 'Hosts' configuration page. The 'Host name' field is 'UbuntuServer-ISE' and the 'Visible name' is 'UbuntuISE'. Under 'Groups', 'Linux servers' is selected. In the 'Interfaces' section, an 'Agent' interface is configured with IP address '192.168.56.105'. The 'Description' field contains the text 'Máquina virtual con Ubuntu Server. Configuración RAID's'. At the bottom, there are checkboxes for 'Monitored by proxy' (set to 'no proxy') and 'Enabled' (checked), along with 'Add' and 'Cancel' buttons.

No debemos olvidar incluir la IP en el fichero de configuración del agente de Zabbix en la máquina con Ubuntu Server como se indica en una nota en el manual de instalación, ya que tenemos Zabbix server en la misma máquina que Zabbix agent. Entonces, modificamos con **vi** */etc/zabbix/zabbix_agentd.conf*

```
### Option: Server
# List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Zabbix servers and Zabbix proxies.
# Incoming connections will be accepted only from the hosts listed here.
# If IPv6 support is enabled then '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' are treated equally
# and '::/0' will allow any IPv4 or IPv6 address.
# '0.0.0.0/0' can be used to allow any IPv4 address.
# Example: Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.56.105
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.56.105
```

Aplicar con: ***systemctl restart zabbix-server zabbix-agent apache2*** y ***systemctl enable zabbix-server zabbix-agent apache2***

Nota: Asegurarse de que los puertos 10050 y 10051 están abiertos con ***ufw allow***.

También debemos añadirle la plantilla para los agentes de Zabbix con sistema operativo Linux que incluye la monitorización de CPU, memoria, sistema de archivos, etc lo que nos permitirá poder ver ciertos gráficos e información. Esta plantilla ya está creada.

Host Templates IPMI Tags Macros **Inventory** Encryption

Linked templates

Name	Action
Template OS Linux by Zabbix agent	Unlink Unlink and clear

Link new templates

Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
UbuntuSE	Applications 19	Items 97	Triggers 35	Graphs 22	Discovery 3	Web	192.168.56.105:10050		Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
Zabbix server	Applications 20	Items 143	Triggers 69	Graphs 28	Discovery 3	Web	192.168.56.105:10050		Template App Zabbix Server, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		

Displaying 2 of 2 found

En esta sección:

Monitoring
Dashboard
Problems
Hosts
Overview
Latest data
Screens
Maps
Discovery
Services
Inventory
Reports

Severity
☐ Not classified
☐ Warning
☐ High
☐ Information
☐ Average
☐ Disaster

Status:
Tags:

Show hosts in maintenance: ☒
Show suppressed problems: ☐

Name	Interface	Availability	Tags	Problems	Status	Latest data	Problems	Graphs	Screens	Web
UbuntuSE	192.168.56.105:10050	ZBX SNMP JMX IPMI			Enabled	Latest data	Problems	Graphs 22	Screens 2	Web
Zabbix server	192.168.56.105:10050	ZBX SNMP JMX IPMI			Enabled	Latest data	Problems	Graphs 28	Screens 3	Web

Displaying 2 of 2 found

Si pulsamos sobre el nombre podemos hacer ping al nuevo host añadido:

Ping

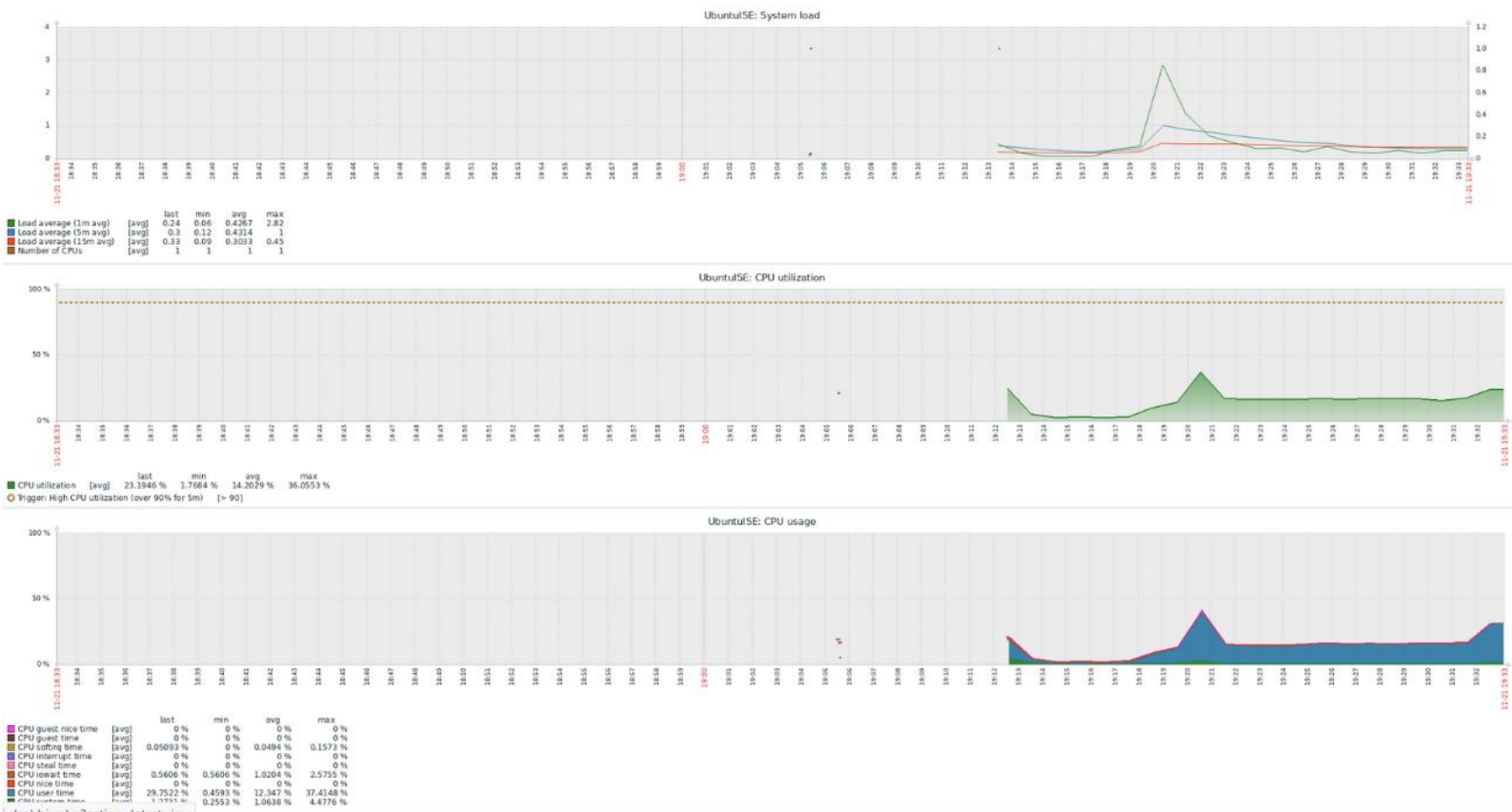
```
ping -c 3 192.168.56.105; case $? in [01]) true;; *) false;; esac

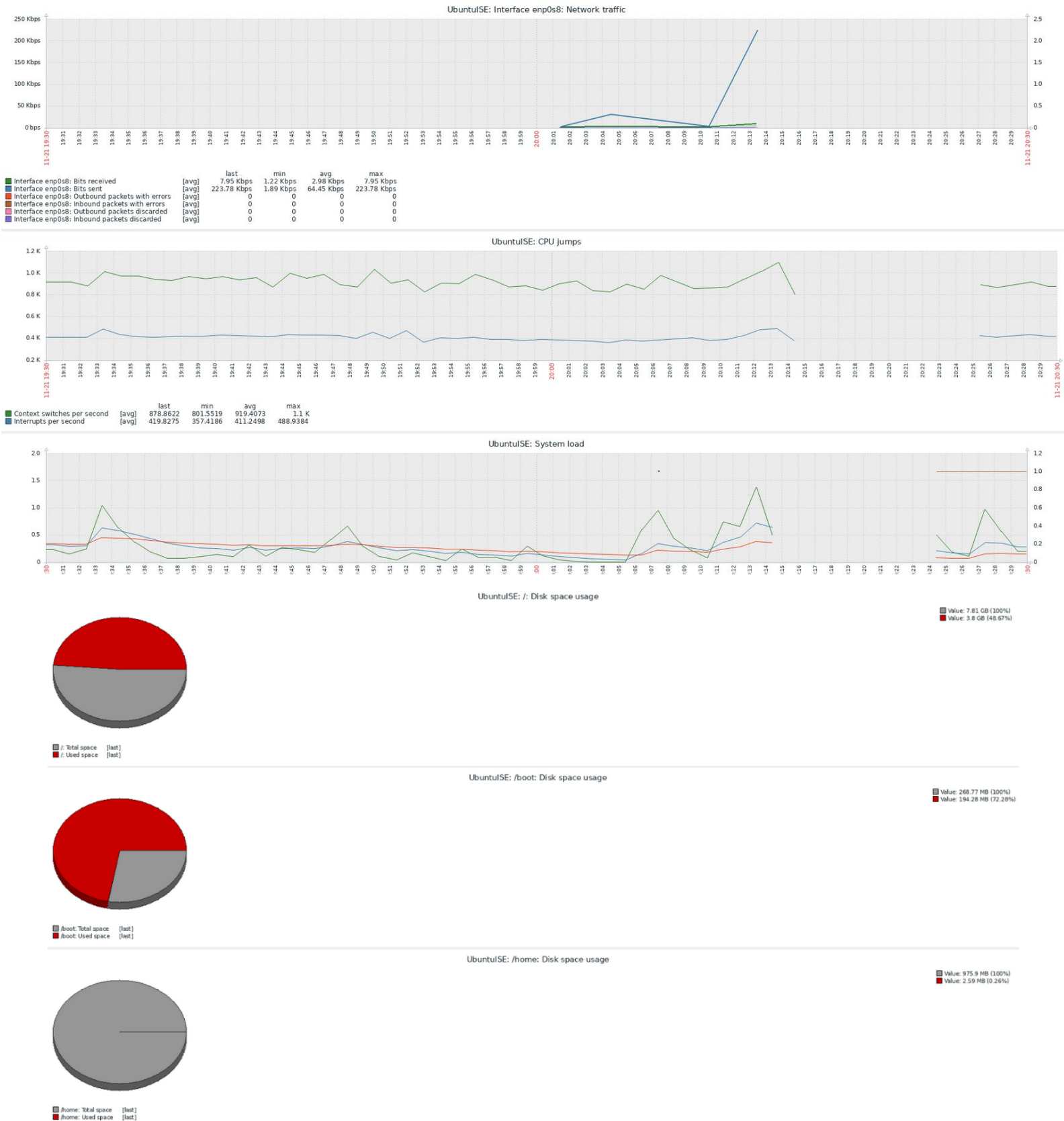
PING 192.168.56.105 (192.168.56.105) 56(84) bytes of data.
64 bytes from 192.168.56.105: icmp_seq=1 ttl=64 time=0.017 ms
64 bytes from 192.168.56.105: icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from 192.168.56.105: icmp_seq=3 ttl=64 time=0.027 ms

--- 192.168.56.105 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.017/0.023/0.027/0.004 ms
```

Cancel

Ya podemos ver ciertas gráficas que se van generando recopilando los datos de nuestra máquina en la sección *Monitoring/Host* y pulsando en la máquina que queramos la opción de *graphs* (estos gráficos también son editables desde la sección de configuración de cada host), por ejemplo, algunas pruebas:





También debemos configurar los *templates* que vienen por defecto con Zabbix para monitorizar los servicios SSH y HTTP. Pero esto lo haremos después de preparar el agente en CentOS.

3. Instalación y Configuración en CentOS - Zabbix Agent:

En este caso debemos instalar únicamente el agente de Zabbix en CentOS. Por tanto, marcamos la versión y seguimos los pasos que nos indica la excepción de los relacionados con la base de datos, el servidor apache o zabbix-server!

https://www.zabbix.com/la/download?zabbix=5.0&os_distribution=centos&os_version=8&db=mysql&ws=apache

VERSIÓN ZABBIX	DISTRIBUCIÓN DE SO	VERSIÓN DEL SISTEMA OPERATIVO	BASE DE DATOS	SERVIDOR WEB
5.2	Red Hat Enterprise Linux	8	MySQL	Apache
5.0 LTS	CentOS	7	PostgreSQL	NGINX
4.0 LTS	Oracle Linux	6		
3.0 LTS	Ubuntu			
	Debian			
	SUSE Linux Enterprise Server			
	Raspbian			

[release notes 5.0](#)

a) Instalar el repositorio de Zabbix con **rpm -Uvh**

https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm y **dnf clean all**

```
CentOS Linux 8 (Core)
Kernel 4.18.0-193.el8.x86_64 on an x86_64

Activate the web console with: systemctl enable --now cockpit.socket

localhost login: peroj
Password:
Last login: Wed Nov 18 16:58:24 on tty1
[peroj@localhost ~]$ sudo su
[sudo] password for peroj:
[root@localhost peroj]# rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
Recuperando https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.noarch.rpm
advertisencia:/var/tmp/rpm-tmp.BZnykw: EncabezadoV4 RSA/SHA512 Signature, ID de clave a14fe591: NOKEY
Verifying... [100%]
Preparando... [100%]
Actualizando / instalando... [100%]
1:zabbix-release-5.0-1.el8
[root@localhost peroj]# dnf clean all
39 archivos eliminados
[root@localhost peroj]#
```

b) Instalar el agente de Zabbix con **dnf install zabbix-agent**

```
[root@localhost peroj]# dnf install zabbix-agent
CentOS-8 - AppStream 3.8 MB/s | 5.8 MB 00:01
CentOS-8 - Base [===] --- B/s | 0 B --:-- ETA

Instalado:
zabbix-agent-5.0.5-1.el8.x86_64

¡Listo!
[root@localhost peroj]#
```

c) Configuración del agente de Zabbix en CentOS:

Modificamos el fichero `/etc/zabbix/zabbix-agentd.conf` como hicimos para configurarlo en Ubuntu, indicando la dirección IP del servidor de Zabbix en los parámetros "Server" y "ActiveServer".

```
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=192.168.56.105
# Mandatory: no
# Default:
# ServerActive=

ServerActive=192.168.56.110
```

Aplicamos los cambios y comprobamos:

```
[root@localhost perojl# systemctl restart zabbix-agent.service
[root@localhost perojl# systemctl enable zabbix-agent.service
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-agent.service → /usr/lib/systemd/system/zabbix-agent.service.
[root@localhost perojl# 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 2020-11-21 14:59:39 EST; 15s ago
 Main PID: 25466 (zabbix_agentd)
    Tasks: 6 (limit: 23960)
   Memory: 3.9M
   CGroup: /system.slice/zabbix-agent.service
           └─25466 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
             └─25467 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
               └─25468 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                 └─25469 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                   └─25470 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                     └─25471 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

nov 21 14:59:39 localhost.localdomain systemd[1]: Starting Zabbix Agent...
nov 21 14:59:39 localhost.localdomain systemd[1]: zabbix-agent.service: Can't open PID file /run/zabbix-agent.pid: Permission denied
nov 21 14:59:39 localhost.localdomain systemd[1]: Started Zabbix Agent.
lines 1-17/17 (END)
```

d) Configuración de los puertos con **firewall-cmd**. Avisamos al cortafuegos de que permita conexiones por ese puerto con **sudo firewall-cmd --permanent --add-port=10050/tcp** y recargamos las reglas con **sudo firewall-cmd --reload**.

```
[root@localhost perojl# firewall-cmd --list-port
22022/tcp 22/tcp
[root@localhost perojl# firewall-cmd --permanent --add-port=10050/tcp
success
[root@localhost perojl# firewall-cmd --reload
success
[root@localhost perojl# firewall-cmd --list-port
22022/tcp 22/tcp 10050/tcp
```


e) Creamos un host nuevo desde el *frontend* de Zabbix para CentOS (agente), como hicimos anteriormente con Ubuntu.

Host Templates IPMI Tags Macros Inventory Encryption

* Host name: CentOS-ISE

Visible name: CentOS-ISE

* Groups: Linux servers

* Interfaces

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

[Add](#)

Description

Monitored by proxy: (no proxy)

Enabled ☒

Añadimos las plantillas (en un principio solo añadimos la misma que hemos añadido en Ubuntu para comprobar que funciona correctamente):

Host Templates IPMI Tags Macros Inventory Encryption

Linked templates

Name	Action
Template OS Linux by Zabbix agent	<input type="button" value="Select"/>

[Add](#) [Cancel](#)

Comprobamos que ya está activo y monitorizando:

Name	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
CentOS-ISE	Applications 11	Items 42	Triggers 14	Graphs 8	Discovery 3	Web	192.168.56.110:10050		Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
UbuntuISE	Applications 19	Items 97	Triggers 35	Graphs 22	Discovery 3	Web	192.168.56.105:10050		Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
Zabbix server	Applications 20	Items 143	Triggers 69	Graphs 28	Discovery 3	Web	192.168.56.105:10050		Template App Zabbix Server, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		

Displaying 3 of 3 found



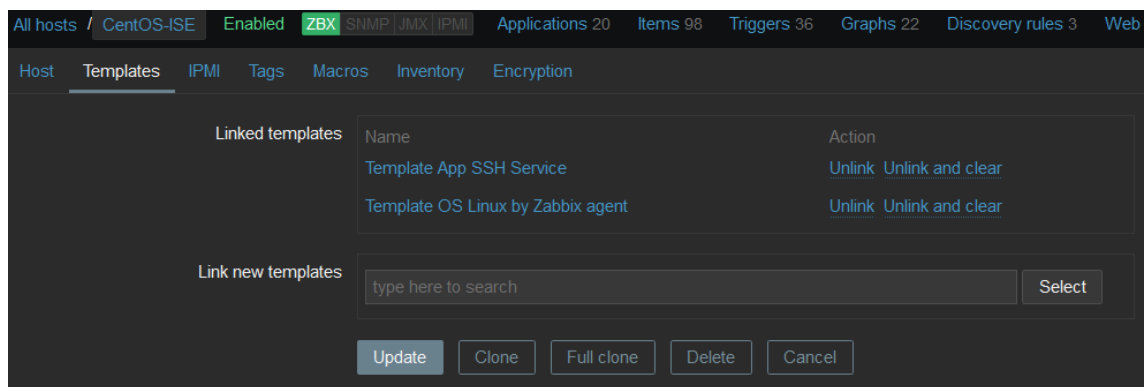
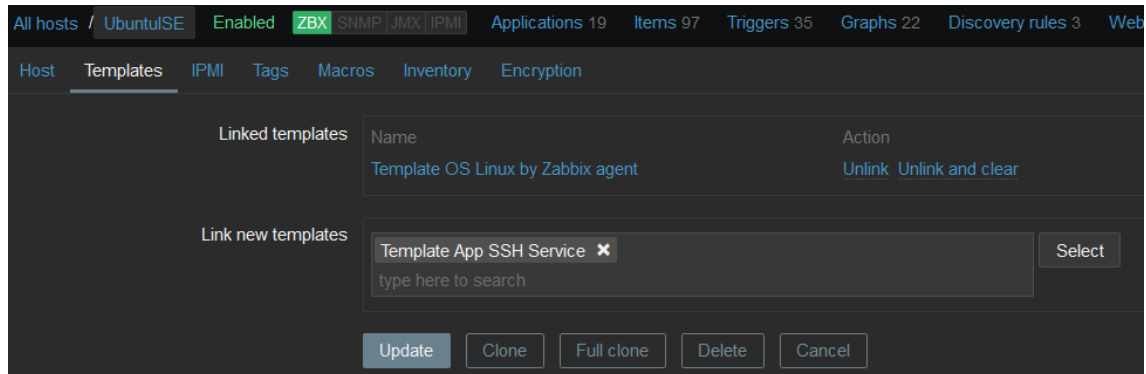
4. Monitorización de servicios SSH:

Consultamos el manual:

<https://www.zabbix.com/documentation/5.0/manual/config/templates>

Tenemos la plantilla *Template App SSH Service* que hará un chequeo simple por el puerto por defecto (22, cambiado previamente).

Para monitorizar el servicio SSH de las dos máquinas debemos irnos al *frontend* y añadirle a cada una de ellas el *template* siguiente:



También se lo añadimos al servidor. Esto creará un nuevo *Trigger* que se disparará cada minuto y controlará si el servicio está activo o no:

<input type="checkbox"/>	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/>	CentOS-ISE	Applications 12	Items 43	Triggers 15	Graphs 8	Discovery 3	Web	192.168.56.110:10050		Template App SSH Service, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
<input type="checkbox"/>	UbuntuSE	Applications 20	Items 98	Triggers 36	Graphs 22	Discovery 3	Web	192.168.56.105:10050		Template App SSH Service, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
<input type="checkbox"/>	Zabbix server	Applications 21	Items 144	Triggers 70	Graphs 28	Discovery 3	Web	192.168.56.105:10050		Template App SSH Service, Template App Zabbix Server, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		

Comprobamos su funcionamiento desactivando el servicio SSH, por ejemplo, en CentOS y vemos como salta el problema:

```
[peroj@localhost ~]$ systemctl stop sshd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Se requiere autenticación para detener 'sshd.service'.
Authenticating as: peroj
Password:
==== AUTHENTICATION COMPLETE ====
[peroj@localhost ~]$
```

Problems

Time ▼	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags
18:22:26		CentOS-ISE	SSH service is down on CentOS-ISE	7m 21s	No		
18:00							

Lo volvemos a activar con **systemctl restart sshd.service** y vemos como se corrige el error:

```
[peroj@localhost ~]$ systemctl restart sshd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Se requiere autenticación para reiniciar 'sshd.service'.
Authenticating as: peroj
Password:
==== AUTHENTICATION COMPLETE ====
[peroj@localhost ~]$ systemctl status sshd.service
• sshd.service - OpenSSH server daemon
  Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2020-11-22 12:31:08 EST; 6s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
  Main PID: 1732 (sshd)
    Tasks: 1 (limit: 23968)
  Memory: 1.1M
  CGroup: /system.slice/sshd.service
          └─1732 /usr/sbin/sshd -D -oCiphers=aes256-gcm@openssh.com,chacha20-poly1305@openssh.com,

nov 22 12:31:08 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
nov 22 12:31:08 localhost.localdomain sshd[1732]: Server listening on 0.0.0.0 port 22.
nov 22 12:31:08 localhost.localdomain sshd[1732]: Server listening on :: port 22.
nov 22 12:31:08 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
```

Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack
18:22:26	Average	18:31:26	RESOLVED		CentOS-ISE	SSH service is down on CentOS-ISE	9m	No

Comprobamos en UbuntuServer:

```
peroj@peroj:~$ sudo systemctl stop sshd.service
peroj@peroj:~$ sudo systemctl status sshd.service
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Sun 2020-11-22 17:35:28 UTC; 42s ago
```

Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
18:38:27	Average				Zabbix server	SSH service is down on Zabbix server	41s	No		
18:38:25	Average				UbuntuISE	SSH service is down on UbuntuISE	43s	No		

Reiniciamos el servicio:

```
UbuntuServerISE (Instantánea 7 - Zabbix Server/Agent (Ubuntu) OK) [Corriendo] - Oracl...
peroj@peroj:~$ sudo systemctl restart sshd.service
peroj@peroj:~$ sudo systemctl status sshd.service
• ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: active (running) since Sun 2020-11-22 17:41:05 UTC; 5s ago
```

Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack
18:38:27	Average	18:41:27	RESOLVED		Zabbix server	SSH service is down on Zabbix server	3m	No
18:38:25	Average	18:41:25	RESOLVED		UbuntuISE	SSH service is down on UbuntuISE	3m	No

5. Monitorización de servicios HTTP:

Pues en este caso usaremos la plantilla *Template App HTTP service* y se la añadimos a los host de igual manera.

<input type="checkbox"/>	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/>	CentOS-ISE	Applications 13	Items 44	Triggers 16	Graphs 8	Discovery 3	Web	192.168.56.110:10050		Template App HTTP Service, Template App SSH Service, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
<input type="checkbox"/>	UbuntuISE	Applications 21	Items 99	Triggers 37	Graphs 22	Discovery 3	Web	192.168.56.105:10050		Template App HTTP Service, Template App SSH Service, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		
<input type="checkbox"/>	Zabbix server	Applications 21	Items 144	Triggers 70	Graphs 28	Discovery 3	Web	192.168.56.105:10050		Template App SSH Service, Template App Zabbix Server, Template OS Linux by Zabbix agent (Template Module Linux block devices by Zabbix agent, Template Module Linux CPU by Zabbix agent, Template Module Linux filesystems by Zabbix agent, Template Module Linux generic by Zabbix agent, Template Module Linux memory by Zabbix agent, Template Module Linux network interfaces by Zabbix agent, Template Module Zabbix agent)	Enabled	ZBX SNMP JMX IPMI	NONE		

Displaying 3 of 3 found

Comprobamos en los últimos datos recibidos por Zabbix y filtramos por nombre:

Latest data

Filter				
Host groups	Linux servers x	Select	Name	http
	type here to search			
Hosts	type here to search	Select	Show items without data	<input checked="" type="checkbox"/>
Application		Select	Show details	<input type="checkbox"/>
		Apply	Reset	
Host	Name ▲	Last check	Last value	Change
CentOS-ISE	HTTP service (1 item)			
<input type="checkbox"/>	HTTP service is running	2020-11-22 18:54:28	Down (0)	Graph
UbuntuISE	HTTP service (1 item)			
<input type="checkbox"/>	HTTP service is running	2020-11-22 18:54:29	Up (1)	Graph

Displaying 2 of 2 found

Time ▼	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack
18:53:28	<input type="checkbox"/> Average		PROBLEM		CentOS-ISE	HTTP service is down on CentOS-ISE	7m 51s	No

Vemos que ya se está monitorizando el servicio en las dos máquinas, pero en CentOS aún nos indica que el ultimo valor fue *Down(0)*. Nos aseguramos que en CentOS esté abierto el puerto 80, utilizado por este servicio por defecto y activado el servicio *httpd*.

```
[root@localhost perojl]# firewall-cmd --list-port
22022/tcp 22/tcp 10050/tcp
[root@localhost perojl]# firewall-cmd --permanent --add-port=80/tcp
success
[root@localhost perojl]# firewall-cmd --list-port
22022/tcp 22/tcp 10050/tcp
[root@localhost perojl]# firewall-cmd --reload
success
[root@localhost perojl]# firewall-cmd --list-port
22022/tcp 22/tcp 10050/tcp 80/tcp
[root@localhost perojl]#

[root@localhost perojl]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
           └─php-fpm.conf
   Active: active (running) since Sun 2020-11-22 13:04:47 EST; 2s ago
     Docs: man:httpd.service(8)
  Main PID: 2118 (httpd)
    Status: "Started, listening on: port 80"
     Tasks: 213 (limit: 23960)
    Memory: 43.0M
   CGroup: /system.slice/httpd.service
           └─2118 /usr/sbin/httpd -DFOREGROUND
             └─2125 /usr/sbin/httpd -DFOREGROUND
               └─2126 /usr/sbin/httpd -DFOREGROUND
                 └─2127 /usr/sbin/httpd -DFOREGROUND
                   └─2128 /usr/sbin/httpd -DFOREGROUND

nov 22 13:04:46 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
nov 22 13:04:47 localhost.localdomain httpd[2118]: AH00558: httpd: Could not reliably determine the
nov 22 13:04:47 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
nov 22 13:04:47 localhost.localdomain httpd[2118]: Server configured, listening on: port 80
lines 1-21/21 (END)
```

Y comprobamos que ahora está en *Up(1)*.

<div><div><div></div></div></div> <div>Host</div>	Name <div></div>	Last check	Last value	Change				
<div><div><div></div></div></div> <div>CentOS-ISE</div>	HTTP service (1 item)							
<div><div><div></div></div></div>	HTTP service is running	2020-11-22 19:05:28	Up (1)	+1 Graph				
<div><div><div></div></div></div> <div>Ubuntu1SE</div>	HTTP service (1 item)							
<div><div><div></div></div></div>	HTTP service is running	2020-11-22 19:05:29	Up (1)	Graph				
Displaying 2 of 2 found								
Time <div></div>	<div><div><div></div></div></div> Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack
18:53:28 <div><div><div></div></div></div>	Average	19:05:28	RESOLVED		CentOS-ISE	HTTP service is down on CentOS-ISE	12m	No

¡Todo correcto!