

Memoria de la Práctica 3

Ingeniería de Servidores

Realizado por:
Juan Valentín Guerrero Cano

Ejercicio 1:

Realice una instalación de Zabbix 5.0 en su servidor con Ubuntu Server20.04 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.

Solución:

Partimos del hecho de que en nuestra máquina UbuntuServer está instalada una pila LAMP en la que en este caso utilizaremos MariaDB en lugar de MySQL.

El primer paso es instalar Zabbix, un sistema de monitorización de redes. En la web oficial de Zabbix podemos localizar los pasos para su correcta instalación.

a. Install Zabbix repository

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

```
valentingc@valentingcubuntu:~$ sudo wget https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-
x-release/zabbix-release_5.0-1+focal_all.deb
--2022-05-11 17:02:54-- https://repo.zabbix.com/zabbix/5.0/ubuntu/pool/main/z/zabbix-release/zabbix
-release_5.0-1+focal_all.deb
Resolving repo.zabbix.com (repo.zabbix.com)... 178.128.6.101, 2604:a880:2:d0::2062:d001
Connecting to repo.zabbix.com (repo.zabbix.com)|178.128.6.101|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4244 (4.1K) [application/octet-stream]
Saving to: 'zabbix-release_5.0-1+focal_all.deb'

zabbix-release_5.0-1+foc 100%[=====] 4.14K --.-KB/s in 0s

2022-05-11 17:02:55 (1.37 GB/s) - 'zabbix-release_5.0-1+focal_all.deb' saved [4244/4244]

valentingc@valentingcubuntu:~$ sudo dpkg -i zabbix-release_5.0-1+focal_all.deb
Selecting previously unselected package zabbix-release.
(Reading database ... 73084 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) ...
valentingc@valentingcubuntu:~$ sudo apt update
Hit:1 http://es.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://es.archive.ubuntu.com/ubuntu focal-updates InRelease
Get:3 http://repo.zabbix.com/zabbix/5.0/ubuntu focal InRelease [4958 B]
Get:4 http://es.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:5 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main Sources [1203 B]
Get:6 http://repo.zabbix.com/zabbix/5.0/ubuntu focal/main amd64 Packages [4785 B]
Get:7 http://es.archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 233 kB in 1s (199 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
221 packages can be upgraded. Run 'apt list --upgradable' to see them.
valentingc@valentingcubuntu:~$
```

b. Install Zabbix server, frontend, agent

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

```
em/zabbix-agent.service.
invoke-rc.d: policy-rc.d denied execution of start.
Setting up libopenipmi0 (2.0.27-0ubuntu2) ...
Setting up fping (4.2-1) ...
Setting up libonig5:amd64 (6.9.4-1) ...
Setting up php-xml (2:7.4+75) ...
Setting up fonts-dejavu (2.37-1) ...
Setting up libsnmp35:amd64 (5.8+dfsg-2ubuntu2.3) ...
Setting up zabbix-server-mysql (1:5.0.23-1+focal) ...
Setting up php-bcmath (2:7.4+75) ...
Setting up php7.4-mbstring (7.4.3-4ubuntu2.10) ...

Creating config file /etc/php/7.4/mods-available/mbstring.ini with new version
Setting up snmpd (5.8+dfsg-2ubuntu2.3) ...
adduser: Warning: The home directory `/var/lib/snmp' does not belong to the user you are currently c
reating.
invoke-rc.d: policy-rc.d denied execution of start.
Created symlink /etc/systemd/system/multi-user.target.wants/snmpd.service → /lib/systemd/system/snmp
d.service.
/usr/sbin/policy-rc.d returned 101, not running 'start snmpd.service'
Setting up php-mbstring (2:7.4+75) ...
Setting up php-ldap (2:7.4+75) ...
Setting up zabbix-frontend-php (1:5.0.23-1+focal) ...
update-alternatives: using /usr/share/fonts/truetype/dejavu/DejaVuSans.ttf to provide /usr/share/zab
bix/assets/fonts/graphfont.ttf (zabbix-frontend-font) in auto mode
Setting up zabbix-apache-conf (1:5.0.23-1+focal) ...
Enabling conf zabbix.
To activate the new configuration, you need to run:
  systemctl reload apache2
Processing triggers for libc-bin (2.31-0ubuntu9) ...
Processing triggers for systemd (245.4-4ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.10) ...
Processing triggers for php7.4-cli (7.4.3-4ubuntu2.10) ...
valentingc@valentingc:~$
```

c. Create initial database

Configuraremos la base de datos obviando al seguridad y usando la contraseña por defecto ya que el proyecto no será expuesto a Internet. Como en nuestro caso trabajamos con MariaDB, tenemos que tener instalado mariadb-server.

```
# mysql -uroot -p
password
mysql> create database zabbix character set utf8 collate utf8_bin;
mysql> create user zabbix@localhost identified by 'password';
mysql> grant all privileges on zabbix.* to zabbix@localhost;
mysql> quit;
```

```
valentingc@valentingcubuntu:~$ sudo mysql -uroot -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 45
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user zabbix@localhost identified by 'practicas,ise';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> quit
Bye
valentingc@valentingcubuntu:~$ _
```

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

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

d. Configure the database for Zabbix server

Editamos el archivo de configuración `etc/zabbix/zabbix_server.conf`:

```
GNU nano 4.8 /etc/zabbix/zabbix_server.conf
### Option: DBSchema
#   Schema name. Used for PostgreSQL.
#
# Mandatory: no
# Default:
# DBSchema=

### Option: DBUser
#   Database user.
#
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

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

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

### Option: DBPort
#   Database port when not using local socket.

[ Wrote 878 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos   M-U Undo
^X Exit      ^R Read File ^_ Replace   ^U Paste Text ^T To Spell  ^_ Go To Line M-E Redo
```

e. Configure PHP for Zabbix frontend

Editamos el archivo etc/zabbix/apache.conf:

```

GNU nano 4.8 /etc/zabbix/apache.conf
# Define /zabbix alias, this is the default
<IfModule mod_alias.c>
    Alias /zabbix /usr/share/zabbix
</IfModule>

<Directory "/usr/share/zabbix">
    Options FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all

    <IfModule mod_php5.c>
        php_value max_execution_time 300
        php_value memory_limit 128M
        php_value post_max_size 16M
        php_value upload_max_filesize 2M
        php_value max_input_time 300
        php_value max_input_vars 10000
        php_value always_populate_raw_post_data -1
        php_value date.timezone Europe/Riga
    </IfModule>
    <IfModule mod_php7.c>
        php_value max_execution_time 300
        php_value memory_limit 128M
        php_value post_max_size 16M
        php_value upload_max_filesize 2M
        php_value max_input_time 300
        php_value max_input_vars 10000
        php_value always_populate_raw_post_data -1
        # php_value date.timezone Europe/Riga
    </IfModule>
</Directory>

[ Wrote 77 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo

```

f. Start Zabbix server and agent processes

Iniciamos el servidor y el agente Zabbix:

```

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

```

```

• zabbix-server.service - Zabbix Server
  Loaded: loaded (/lib/systemd/system/zabbix-server.service; disabled; vendor preset: enabled)
  Active: active (running) since Wed 2022-05-11 17:24:01 UTC; 4min 1s ago
  Main PID: 14929 (zabbix_server)
  Tasks: 38 (limit: 2274)
  Memory: 29.8M
  CGroup: /system.slice/zabbix-server.service
          └─14929 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
          └─14939 /usr/sbin/zabbix_server: configuration syncer [synced configuration in 0.00613s]
          └─14940 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.003s]
          └─14941 /usr/sbin/zabbix_server: alerter #1 started
          └─14942 /usr/sbin/zabbix_server: alerter #2 started
          └─14943 /usr/sbin/zabbix_server: alerter #3 started
          └─14944 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 0, processed 5 values]
          └─14945 /usr/sbin/zabbix_server: preprocessing worker #1 started
          └─14946 /usr/sbin/zabbix_server: preprocessing worker #2 started
          └─14947 /usr/sbin/zabbix_server: preprocessing worker #3 started
          └─14948 /usr/sbin/zabbix_server: lld manager #1 [processed 0 LLD rules, idle 5.020172s]
          └─14949 /usr/sbin/zabbix_server: lld worker #1 started
          └─14950 /usr/sbin/zabbix_server: lld worker #2 started
          └─14951 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes]
          └─14952 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.0s]
          └─14953 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.000479 sec, idle 5s]
          └─14954 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.000483 sec, idle 1s]
          └─14955 /usr/sbin/zabbix_server: history syncer #1 [processed 0 values, 0 triggers in 0.0s]
          └─14956 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0 triggers in 0.0s]
          └─14957 /usr/sbin/zabbix_server: history syncer #3 [processed 0 values, 0 triggers in 0.0s]
          └─14971 /usr/sbin/zabbix_server: history syncer #4 [processed 0 values, 0 triggers in 0.0s]
          └─14972 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.000553 sec]
          └─14973 /usr/sbin/zabbix_server: proxy poller #1 [exchanged data with 0 proxies in 0.0s]
          └─14977 /usr/sbin/zabbix_server: self-monitoring [processed data in 0.000024 sec, idle 1s]
          └─14978 /usr/sbin/zabbix_server: task manager [processed 0 task(s) in 0.000191 sec, idle 1s]
          └─14979 /usr/sbin/zabbix_server: poller #1 [got 1 values in 0.000299 sec, idle 1 sec]
          └─14982 /usr/sbin/zabbix_server: poller #2 [got 0 values in 0.000005 sec, idle 1 sec]
          └─14983 /usr/sbin/zabbix_server: poller #3 [got 0 values in 0.000007 sec, idle 1 sec]
          └─14985 /usr/sbin/zabbix_server: poller #4 [got 0 values in 0.000005 sec, idle 1 sec]
lines 1-36

```

```

valentingc@valentingcubuntu:~$ systemctl status zabbix-agent
• zabbix-agent.service - Zabbix Agent
  Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2022-05-11 17:24:01 UTC; 4min 56s ago
  Main PID: 14926 (zabbix_agentd)
  Tasks: 6 (limit: 2274)
  Memory: 6.0M
  CGroup: /system.slice/zabbix-agent.service
          └─14926 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
          └─14932 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
          └─14933 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
          └─14934 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
          └─14935 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
          └─14936 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

May 11 17:24:01 valentingcubuntu systemd[1]: Starting Zabbix Agent...
May 11 17:24:01 valentingcubuntu systemd[1]: Started Zabbix Agent.
valentingc@valentingcubuntu:~$

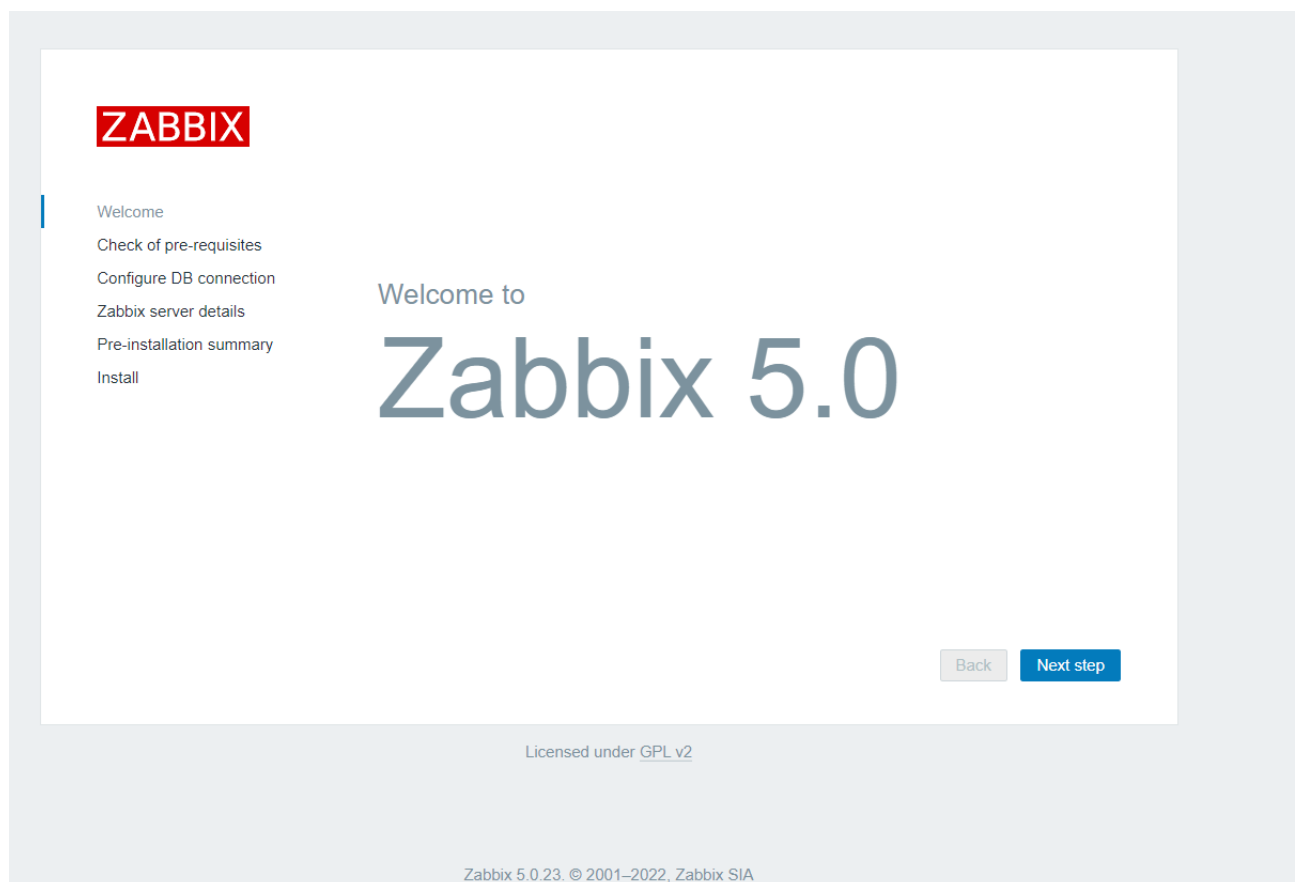
```

```
valentingc@valentingcubuntu:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-05-11 17:24:01 UTC; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 15005 (apache2)
    Tasks: 9 (limit: 2274)
   Memory: 22.0M
    CGroup: /system.slice/apache2.service
            └─15005 /usr/sbin/apache2 -k start
              └─15006 /usr/sbin/apache2 -k start
                └─15007 /usr/sbin/apache2 -k start
                  └─15008 /usr/sbin/apache2 -k start
                    └─15009 /usr/sbin/apache2 -k start
                      └─15010 /usr/sbin/apache2 -k start
                        └─15303 /usr/sbin/apache2 -k start
                          └─15304 /usr/sbin/apache2 -k start
                            └─15305 /usr/sbin/apache2 -k start

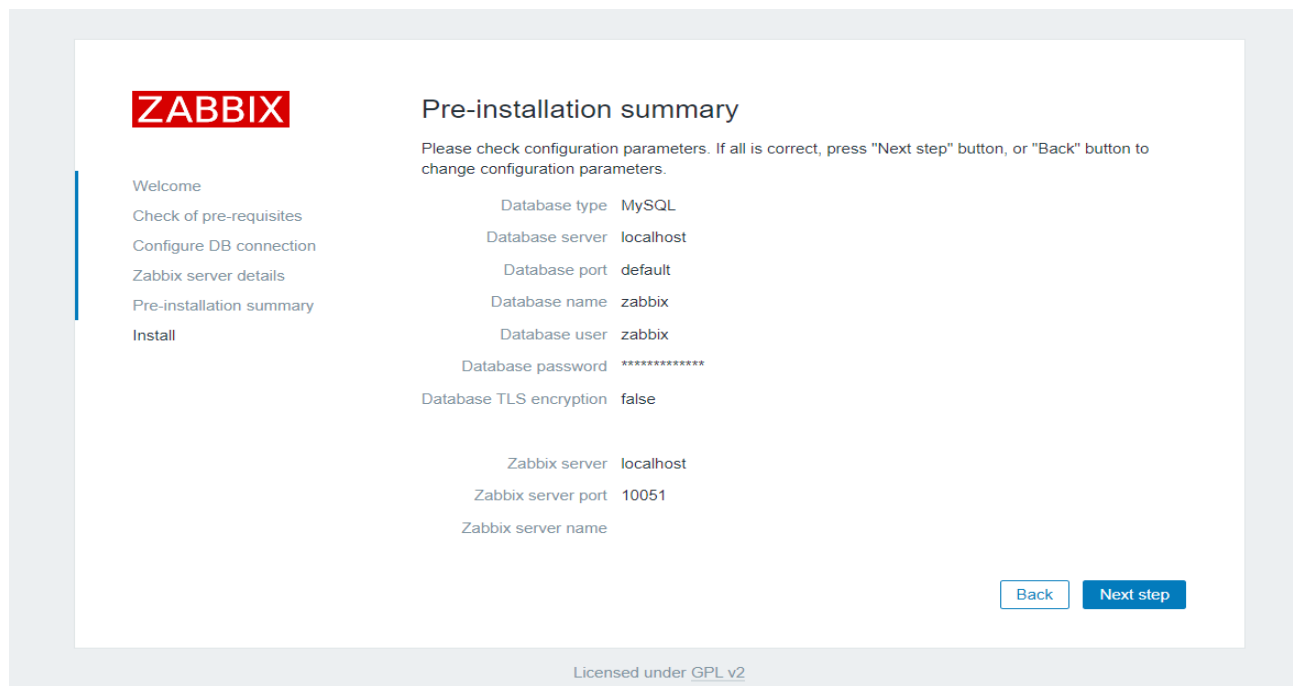
May 11 17:24:01 valentingcubuntu systemd[1]: Starting The Apache HTTP Server...
May 11 17:24:01 valentingcubuntu apache2ctl[14963]: AH00558: apache2: Could not reliably determine t
May 11 17:24:01 valentingcubuntu systemd[1]: Started The Apache HTTP Server.
```

g. Configure Zabbix frontend

Acabamos de instalar nuestro servidor. Accedamos al frontend del mismo
<http://192.168.56.105/zabbix>



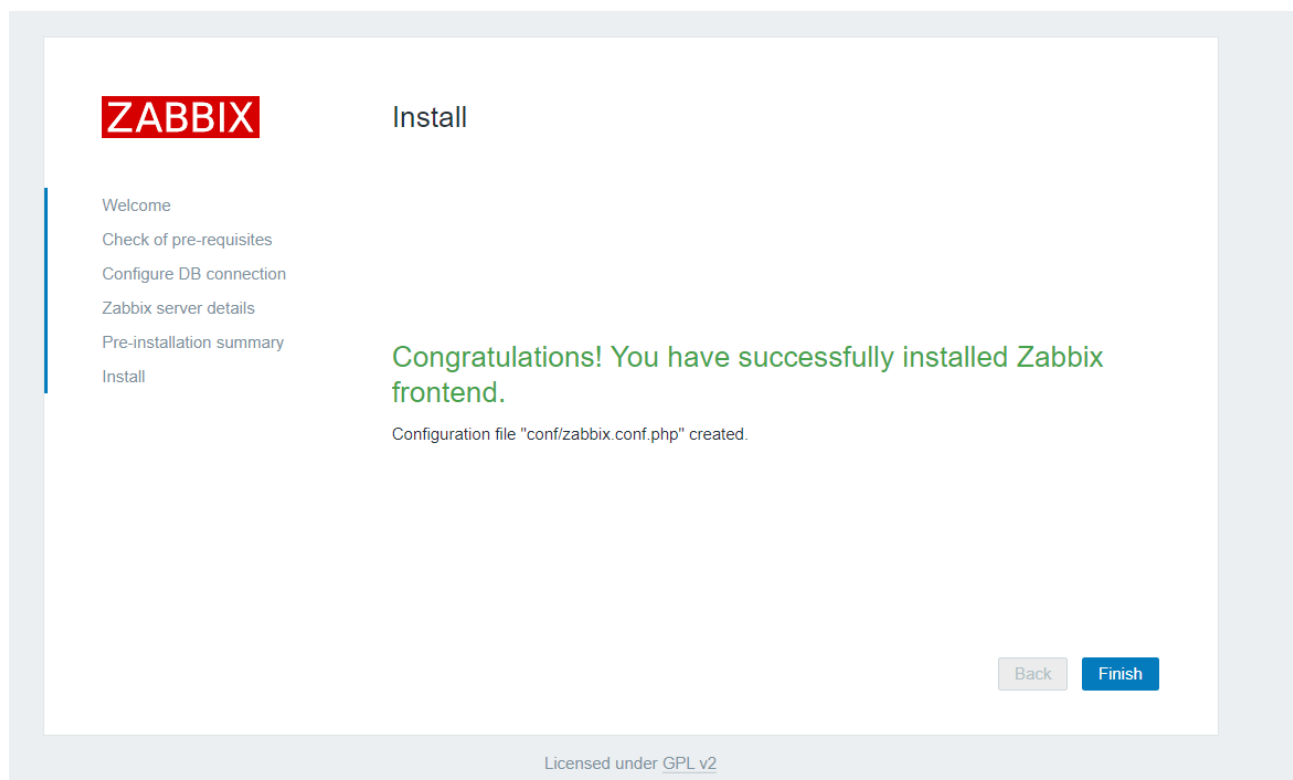
Procedemos a terminar la configuración.



The screenshot shows the Zabbix Pre-installation summary screen. On the left is a sidebar with the ZABBIX logo and a list of steps: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details, Pre-installation summary (highlighted), and Install. The main area is titled 'Pre-installation summary' and contains a list of configuration parameters with their values. At the bottom right are 'Back' and 'Next step' buttons. The footer indicates the software is licensed under GPL v2.

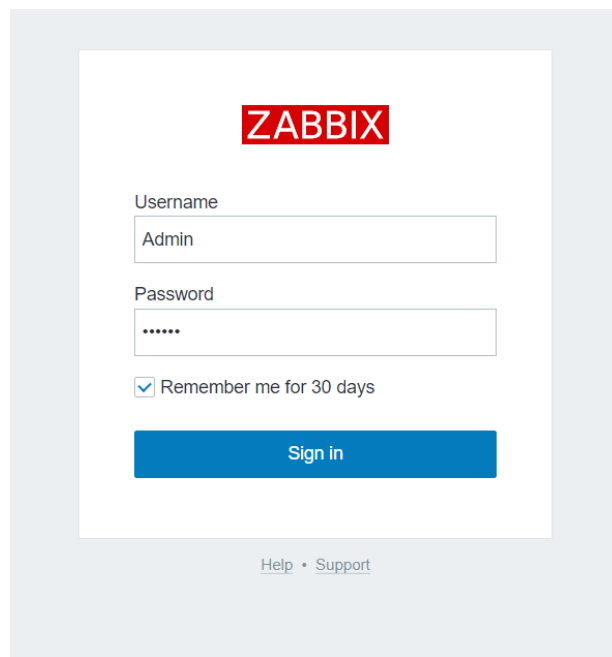
Parameter	Value
Database type	MySQL
Database server	localhost
Database port	default
Database name	zabbix
Database user	zabbix
Database password	*****
Database TLS encryption	false
Zabbix server	localhost
Zabbix server port	10051
Zabbix server name	

Comprobamos todos los requisitos previos a la instalación.



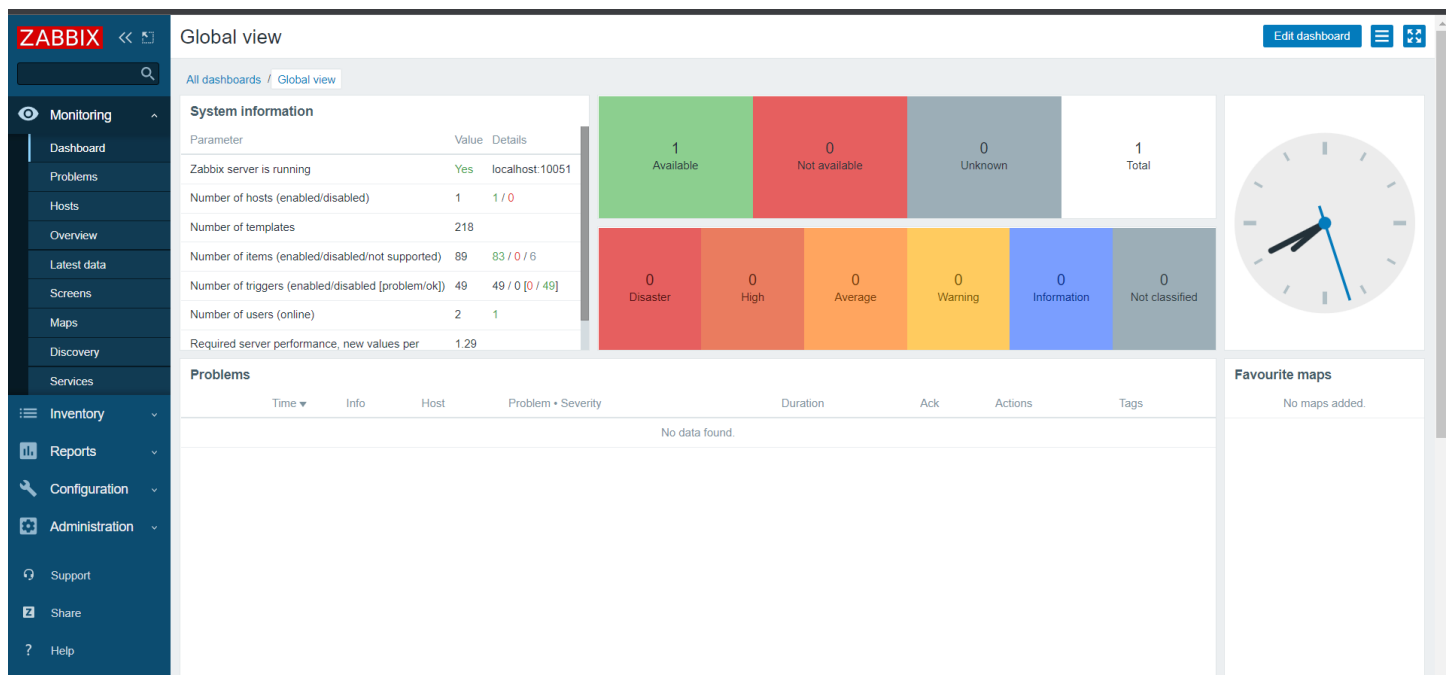
The screenshot shows the Zabbix Install screen. The sidebar is identical to the previous screen, with 'Install' highlighted. The main area is titled 'Install' and displays a green message: 'Congratulations! You have successfully installed Zabbix frontend.' Below this, it states 'Configuration file "conf/zabbix.conf.php" created.' At the bottom right are 'Back' and 'Finish' buttons. The footer indicates the software is licensed under GPL v2.

En vistas a la documentación seguida, el nuevo usuario para acceder al frontend de Zabbix es Admin con contraseña zabbix tal y como se puede observar en la captura.



The image shows the Zabbix login interface. It features a white login box centered on a light gray background. At the top of the box is the ZABBIX logo in a red rectangle. Below the logo are two input fields: 'Username' with 'Admin' entered, and 'Password' with masked characters. A checkbox labeled 'Remember me for 30 days' is checked. A blue 'Sign in' button is positioned below the password field. At the bottom of the login box, there are links for 'Help' and 'Support'.

Si nos logueamos con los datos previamente dichos nos encontraremos con:



The image displays the Zabbix 'Global view' dashboard. On the left is a dark blue sidebar with navigation links: Monitoring, Inventory, Reports, Configuration, Administration, Support, Share, and Help. The main content area is titled 'Global view' and includes a search bar and an 'Edit dashboard' button. The dashboard is divided into several sections:

- System information:** A table with columns 'Parameter', 'Value', and 'Details'.

Parameter	Value	Details
Zabbix server is running	Yes	localhost:10051
Number of hosts (enabled/disabled)	1	1 / 0
Number of templates	218	
Number of items (enabled/disabled/not supported)	89	83 / 0 / 6
Number of triggers (enabled/disabled [problem/ok])	49	49 / 0 [0 / 49]
Number of users (online)	2	1
Required server performance, new values per	1.29	
- Status Overview:** A row of colored boxes showing counts: 1 Available (green), 0 Not available (red), 0 Unknown (gray), and 1 Total (white). Below this is another row: 0 Disaster (red), 0 High (orange), 0 Average (yellow), 0 Warning (yellow), 0 Information (blue), and 0 Not classified (gray).
- Problems:** A table with columns: Time, Info, Host, Problem • Severity, Duration, Ack, Actions, and Tags. It currently shows 'No data found.'
- Other Widgets:** A clock on the right, and a 'Favourite maps' section at the bottom right stating 'No maps added.'

Hemos finalizado la instalación de zabbix. A continuación instalaremos un agente en CentOS para poder monitorizarlo también. Primeros procederemos a desactivar SELinux para evitar futuros problemas. Accedemos al fichero de configuración: etc/selinux/config y modificamos la variable SELINUX=disabled tal y como se muestra en la imagen.

```

GNU nano 2.9.8 /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

```

A continuación instalaremos el repositorio 5.0 de Zabbix, e instalaremos zabbix-agent tal y como se muestra en las capturas.

```

[valentingc@localhost yum.repos.d]$ sudo dnf install https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_6
4/zabbix-release-5.0-1.el8.noarch.rpm
Última comprobación de caducidad de metadatos hecha hace 0:00:50, el mié 11 may 2022 14:10:05 EDT.
zabbix-release-5.0-1.el8.noarch.rpm 17 kB/s | 18 kB 00:01
Dependencias resueltas.
=====
Paquete Arquitectura Versión Repositorio Tam.
=====
Instalando:
zabbix-release noarch 5.0-1.el8 @commandline 18 k
Resumen de la transacción
=====
Instalar 1 Paquete

Tamaño total: 18 k
Tamaño instalado: 21 k
¿Está de acuerdo [s/N]? : s
Descargando paquetes:
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
Preparando : 1/1
Instalando : zabbix-release-5.0-1.el8.noarch 1/1
Verificando : zabbix-release-5.0-1.el8.noarch 1/1

Instalado:
zabbix-release-5.0-1.el8.noarch

¡Listo!
[valentingc@localhost yum.repos.d]$ _

```

```

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

Tamaño total de la descarga: 492 k
Tamaño instalado: 2.1 M
¿Está de acuerdo [s/N]? : s
Descargando paquetes:
zabbix-agent-5.0.23-1.el8.x86_64.rpm                216 kB/s | 492 kB      00:02
-----
Total                                              216 kB/s | 492 kB      00:02
advertencia:/var/cache/dnf/zabbix-b7349cbb4866b08d/packages/zabbix-agent-5.0.23-1.el8.x86_64.rpm: Encabezado V4 RSA/SHA512 Signature, ID de clave a14fe591: NOKEY
Zabbix Official Repository - x86_64                1.6 MB/s | 1.7 kB      00:00
Importando llave GPG 0xA14FE591:
  ID usuario: "Zabbix LLC <packager@zabbix.com>"
  Huella      : A184 8F53 52D0 22B9 471D 83D0 082A B56B A14F E591
  Desde       : /etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
¿Está de acuerdo [s/N]? : s
La llave ha sido importada exitosamente
Ejecutando verificación de operación
Verificación de operación exitosa.
Ejecutando prueba de operaciones
Prueba de operación exitosa.
Ejecutando operación
  Preparando          :                               1/1
  Ejecutando scriptlet: zabbix-agent-5.0.23-1.el8.x86_64 1/1
  Instalando          : zabbix-agent-5.0.23-1.el8.x86_64 1/1
  Ejecutando scriptlet: zabbix-agent-5.0.23-1.el8.x86_64 1/1
  Verificando         : zabbix-agent-5.0.23-1.el8.x86_64 1/1

Instalado:
  zabbix-agent-5.0.23-1.el8.x86_64

¡Listo!
[valentingc@localhost yum.repos.d]$_

```

Comprobamos que podemos iniciar el servicio de forma correcta:

```

[valentingc@localhost ~]$ systemctl status zabbix-agent
● zabbix-agent.service - Zabbix Agent
   Loaded: loaded (/usr/lib/systemd/system/zabbix-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-05-11 14:17:36 EDT; 3s ago
     Process: 22876 ExecStart=/usr/sbin/zabbix_agentd -c $CONFFILE (code=exited, status=0/SUCCESS)
    Main PID: 22878 (zabbix_agentd)
      Tasks: 6 (limit: 11472)
     Memory: 3.8M
    CGroup: /system.slice/zabbix-agent.service
            └─22878 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
              └─22879 /usr/sbin/zabbix_agentd: collector [idle 1 sec]
                └─22880 /usr/sbin/zabbix_agentd: listener #1 [waiting for connection]
                  └─22881 /usr/sbin/zabbix_agentd: listener #2 [waiting for connection]
                    └─22882 /usr/sbin/zabbix_agentd: listener #3 [waiting for connection]
                      └─22883 /usr/sbin/zabbix_agentd: active checks #1 [idle 1 sec]

may 11 14:17:36 localhost.localdomain systemd[1]: Starting Zabbix Agent...
may 11 14:17:36 localhost.localdomain systemd[1]: Started Zabbix Agent.
[valentingc@localhost ~]$ _

```

Ahora modificamos los parámetros como deseamos a través del archivo de configuración: etc/zabbix/zabbix_agentd.conf.

```

GNU nano 2.9.8                               /etc/zabbix/zabbix_agentd.conf      Modi
# List of comma delimited IP addresses, optionally in CIDR notation, or DNS names of Z
# 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 t
# 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=192.168.56.105,127.0.0.1

```

Una vez realizamos los cambios, recargamos el servicio. (systemctl restart zabbix-agent). Hay que tener en cuenta que tenemos que habilitar los respectivos puertos y servicios para permitir la transmisión de datos. Para ello :

```

ServerActive=192.168.56.105,127.0.0.1

### Option: Hostname
# Unique, case sensitive hostname.
# Required for active checks and must match hostname as configured on the server.
# Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=
Hostname=CentOS

```

```

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

```

Por último, una vez iniciado el servicio zabbix-server tenemos que configurar en el frontend de Zabbix los hosts que monitorizamos. Concretamente tenemos que crear uno nuevo con los datos correctos tal y como se muestra en la imagen.

<input type="checkbox"/>	Name ▲	Applications	Items	Triggers	Graphs	Discovery	Web	Interface	Proxy	Templates	Status	Availability	Agent encryption	Info	Tags
<input type="checkbox"/>	CentOS	Applications	Items	Triggers	Graphs	Discovery	Web	192.168.56.110:10050			Enabled	ZBX SNMP JMX IPMI	NONE		
<input type="checkbox"/>	Zabbix server	Applications 17	Items 124	Triggers 66	Graphs 25	Discovery 3	Web	127.0.0.1: 10050			Enabled	ZBX SNMP JMX IPMI	NONE		

A continuación tenemos que crear nuevos items con los parámetros adecuados para poder monitorizar lo que nos pide el enunciado en el host recién creado:

<input type="checkbox"/>	Wizard	Name ▲	Triggers	Key	Interval	History	Trends	Type	Applications	Status	Info
<input type="checkbox"/>	***	CPU load monitoring		system.cpu.load	1m	90d	365d	Zabbix agent		Enabled	
<input type="checkbox"/>	***	escritura en disco monitoring		vfs.dev.write	1m	90d	365d	Zabbix agent		Enabled	
<input type="checkbox"/>	***	http monitoring		net.tcp.service[http]	1m	90d	365d	Zabbix agent		Enabled	
<input type="checkbox"/>	***	lectura en disco monitoring		vfs.dev.read	1m	90d	365d	Zabbix agent		Enabled	
<input type="checkbox"/>	***	ssh monitoring		net.tcp.service[ssh]	1m	90d	365d	Zabbix agent		Enabled	

Displaying 5 of 5 found

Los parámetros para crear cada uno de los items se pueden encontrar en la documentación oficial.

Podemos monitorizar cada uno de los parámetros a través de la pestaña “Monitoring” del frontend de Zabbix.

ZABBIX

<<>>

Filter

Monitoring

Dashboard

Problems

Hosts

Overview

Latest data

Screens

Maps

Discovery

Services

Inventory

Reports

Configuration

Administration

Support

Share

Help

Latest data

Host groups: type here to search Select

Hosts: type here to search Select

Application: Select

Name:

Show items without data: ☒

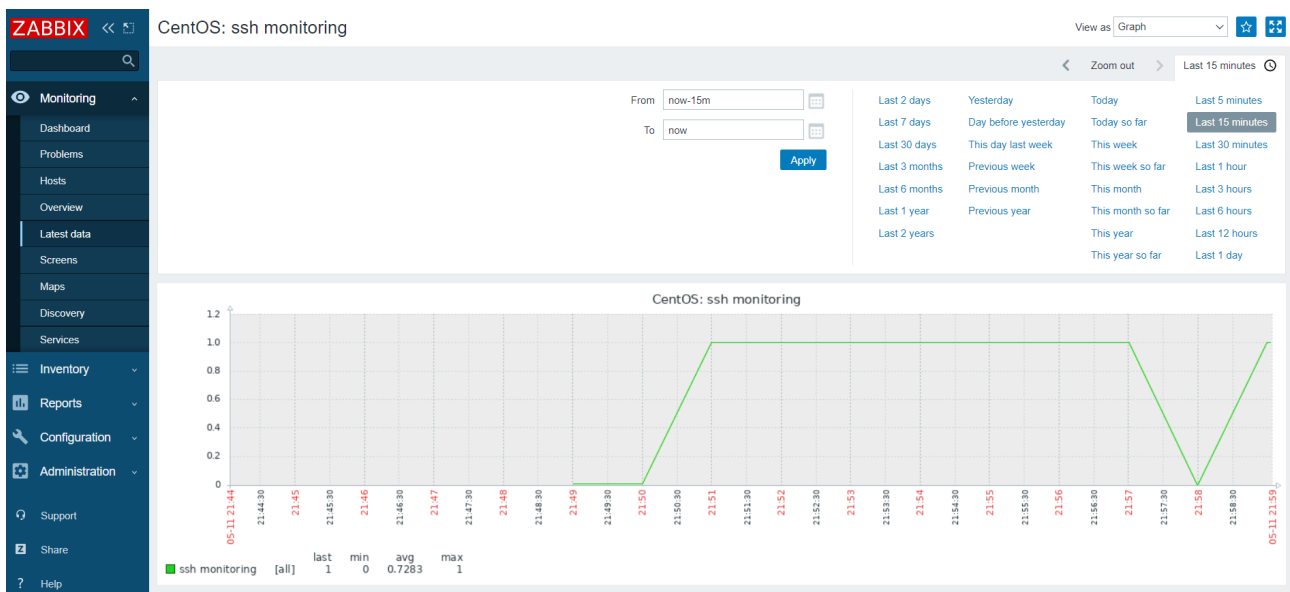
Show details: ☐

Apply

Reset

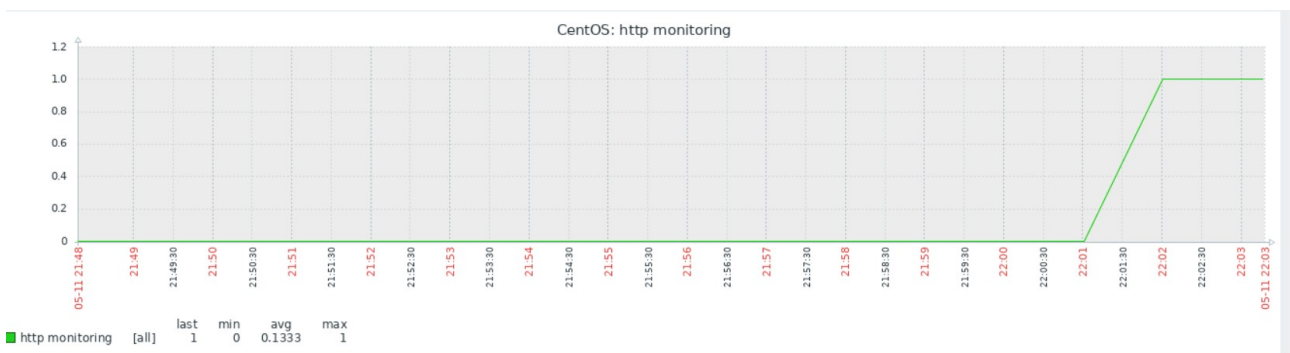
<input type="checkbox"/>	Host	Name ▲	Last check	Last value	Change
<input type="checkbox"/>	CentOS	- other - (5 Items)			
<input type="checkbox"/>		CPU load monitoring	2022-05-11 21:53:03	0	Graph
<input type="checkbox"/>		escritura en disco monitoring	2022-05-11 21:52:04	2.7333	-0.4 Graph
<input type="checkbox"/>		http monitoring	2022-05-11 21:53:02	0	Graph
<input type="checkbox"/>		lectura en disco monitoring	2022-05-11 21:52:05	0	Graph
<input type="checkbox"/>		ssh monitoring	2022-05-11 21:53:01	1	Graph
<input type="checkbox"/>	Zabbix server	CPU (17 Items)			
<input type="checkbox"/>		Context switches per second	2022-05-11 21:53:03	517.8967	+118.6993 Graph
<input type="checkbox"/>		CPU guest nice time	2022-05-11 21:52:04	0 %	Graph
<input type="checkbox"/>		CPU guest time	2022-05-11 21:52:05	0 %	Graph
<input type="checkbox"/>		CPU idle time	2022-05-11 21:52:13	98.454 %	+0.05041 % Graph
<input type="checkbox"/>		CPU interrupt time	2022-05-11 21:52:07	0 %	Graph
<input type="checkbox"/>		CPU iowait time	2022-05-11 21:53:02	0.9405 %	+0.04949 % Graph
<input type="checkbox"/>		CPU nice time	2022-05-11 21:52:09	0 %	Graph

Si comprobamos por ejemplo la gráfica del servicio ssh en CentOS, y probamos a parar y reactivar el servicio podemos observar cómo varía:

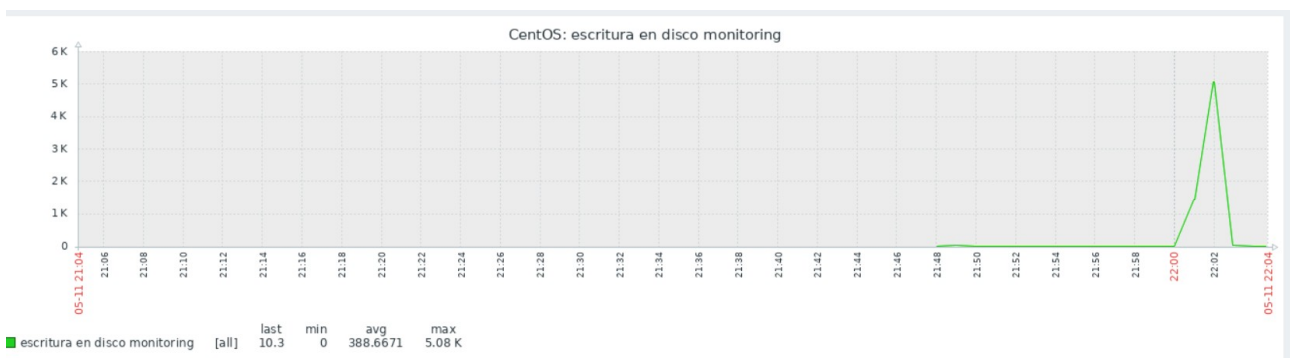


Tomando otras gráficas como ejemplo:

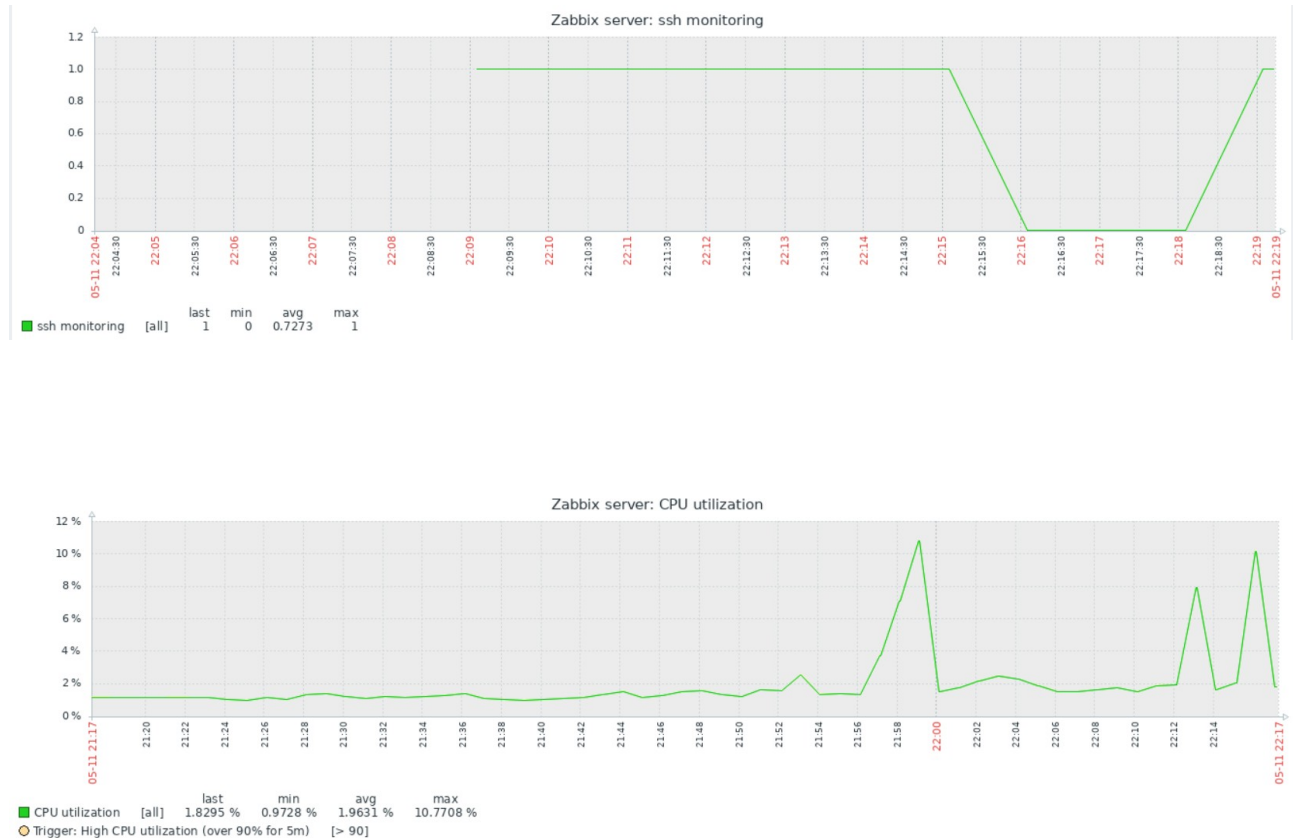
- La monitorización del servicio http (httpd en CentOS):



- La monitorización de la escritura en disco:



Una vez terminada la monitorización de CentOS continuemos con la monitorización sobre Zabbix server de la MV UbuntuServer. Esta es análoga a la de CentOS, adjuntamos algunas capturas que muestran el proceso para la monitorización del servicio ssh y de la utilización de CPU como ejemplos.



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.

Solución:

Realizaremos la instalación de Ansible en Ubuntu Server siguiendo la documentación oficial para guiarnos. Cabe remarcar que el Ansible no necesita agentes en el lado del cliente, por lo que solo tendremos que configurar el servidor.

En la documentación oficial nos menciona que hay nodos controladores desde los cuales mediante Ansible, se pueden controlar otros dispositivos, muy importante, mediante el protocolo SSH (por defecto).

Como requerimientos para los nodos controladores (en nuestro caso Ubuntu Server) nos pide que tengamos como mínimo la versión 3.8 de Python.

```
valentingc@valentingcubuntu:~$ sudo apt install python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
0 upgraded, 0 newly installed, 0 to remove and 187 not upgraded.
valentingc@valentingcubuntu:~$ sudo apt install python-is-python3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  python-is-python3
0 upgraded, 1 newly installed, 0 to remove and 187 not upgraded.
Need to get 2364 B of archives.
After this operation, 10.2 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu focal/main amd64 python-is-python3 all 3.8.2-4 [2364 B]
Fetched 2364 B in 0s (15.7 kB/s)
Selecting previously unselected package python-is-python3.
(Reading database ... 74971 files and directories currently installed.)
Preparing to unpack .../python-is-python3_3.8.2-4_all.deb ...
Unpacking python-is-python3 (3.8.2-4) ...
Setting up python-is-python3 (3.8.2-4) ...
valentingc@valentingcubuntu:~$
```

Ya lo teníamos instalado. Continuamos con la documentación, ahora toca instalar Ansible.

```

Preparing to unpack .../4-python3-requests-kerberos_0.12.0-2_all.deb ...
Unpacking python3-requests-kerberos (0.12.0-2) ...
Selecting previously unselected package python3-requests-ntlm.
Preparing to unpack .../5-python3-requests-ntlm_1.1.0-1_all.deb ...
Unpacking python3-requests-ntlm (1.1.0-1) ...
Selecting previously unselected package python3-xlrd.
Preparing to unpack .../6-python3-xlrd_0.12.0-1_all.deb ...
Unpacking python3-xlrd (0.12.0-1) ...
Selecting previously unselected package python3-winrm.
Preparing to unpack .../7-python3-winrm_0.3.0-2_all.deb ...
Unpacking python3-winrm (0.3.0-2) ...
Setting up python3-ntlm-auth (1.1.0-1) ...
Setting up python3-kerberos (1.1.14-3.1build1) ...
Setting up python3-xlrd (0.12.0-1) ...
Setting up python3-jmespath (0.9.4-2) ...
/usr/lib/python3/dist-packages/jmespath/visitor.py:32: SyntaxWarning: "is" with a literal. Did you mean "=="?
    if x is 0 or x is 1:
/usr/lib/python3/dist-packages/jmespath/visitor.py:32: SyntaxWarning: "is" with a literal. Did you mean "=="?
    if x is 0 or x is 1:
/usr/lib/python3/dist-packages/jmespath/visitor.py:34: SyntaxWarning: "is" with a literal. Did you mean "=="?
    elif y is 0 or y is 1:
/usr/lib/python3/dist-packages/jmespath/visitor.py:34: SyntaxWarning: "is" with a literal. Did you mean "=="?
    elif y is 0 or y is 1:
/usr/lib/python3/dist-packages/jmespath/visitor.py:260: SyntaxWarning: "is" with a literal. Did you mean "=="?
    if original_result is 0:
Setting up python3-requests-kerberos (0.12.0-2) ...
Setting up ansible (5.7.1-1ppa~focal) ...
Setting up python3-requests-ntlm (1.1.0-1) ...
Setting up python3-winrm (0.3.0-2) ...
valentingc@valentingcubuntu:~$

```

Comprobemos que se ha realizado correctamente. Para ello utilizaremos el comando que se muestra en la consola:

```

valentingc@valentingcubuntu:~$ ansible all -m ping -u valentingc
[WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'
valentingc@valentingcubuntu:~$

```

Aunque nos de un error esto nos muestra que finalmente la instalación se ha realizado correctamente.

A continuación procedamos a configurar Ansible de la forma en la que deseamos. A través del archivo de configuración de Ansible (`etc/ansible/ansible.cfg`) podemos realizar las modificaciones requeridas. Lo primero es realizar el “inventario” que en Ansible se conoce como el conjunto de máquinas que queremos controlar, que en nuestro caso es sencillamente CentOS (192.168.56.110) y la IP local (127.0.0.1). Para ello tenemos que modificar el archivo `hosts`:

```
GNU nano 4.8 /etc/ansible/hosts
## 192.168.100.1
## 192.168.100.10

# Ex 2: A collection of hosts belonging to the 'webservers' group:

## [webservers]
## alpha.example.org
## beta.example.org
## 192.168.1.100
## 192.168.1.110

# If you have multiple hosts following a pattern, you can specify
# them like this:

## www[001:006].example.com

# Ex 3: A collection of database servers in the 'dbservers' group:

## [dbservers]
##
## db01.intranet.mydomain.net
## db02.intranet.mydomain.net
## 10.25.1.56
## 10.25.1.57

# Here's another example of host ranges, this time there are no
# leading 0s:

## db-[99:101]-node.example.com
[test]
192.168.56.110
127.0.0.1

^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos    M-U Undo
^X Exit      ^R Read File  ^_ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo
```

Los agruparemos bajo un alias, así en las pruebas que realicemos a continuación se llamarán conjuntamente a ambas máquinas y no tendremos que realizar las pruebas máquina a máquina. Como estas máquinas virtuales son nuevas y no las de prácticas anteriores el puerto de ssh es el 22. Por lo que si nos encontrásemos con que hemos cambiado el puerto de ssh deberíamos modificar el ya mencionado archivo de configuración de Ansible. También hay que tener correctamente configuradas las claves ssh para poder conectarnos adecuadamente. Probemos a ordenar un ping desde el nodo control a los controlados:

```

valentingc@valentingcubuntu:~$ ansible all -m ping
The authenticity of host '127.0.0.1 (127.0.0.1)' can't be established.
ECDSA key fingerprint is SHA256:Uui0mMDMGTKyNGaUJcSj2DoeUb28fEvxkVb7W9J96+I.
Are you sure you want to continue connecting (yes/no/[fingerprint])? 192.168.56.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
yes
127.0.0.1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
valentingc@valentingcubuntu:~$ _

```

Como bien describe el enunciado existe la posibilidad de automatizar procesos y tareas mediante playbooks. Podemos obtener también privilegios en otras máquinas mediante la opción “become” de Ansible. Esta opción nos permitiría ejecutar la orden en la máquina como sudo. Ahora creemos un script sencillo que podamos automatizar con Ansible y ejecutarlo en todas las máquinas (Ubuntu y CentOS) con una orden:

```

GNU nano 4.8                                playbook.yml
= name: Mostrar por consola
  hosts: all
  remote_user: valentingc
  tasks:
    - name: Ejecutar el script
      script: /home/valentingc/script.sh

```

[Read 6 lines]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
 ^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo

Para crear el playbook simplemente tenemos que crear un archivo con el formato .yml y hay que guardad correctamente la indentación del código a utilizar ya que he experimentado varios errores simplemente por no guardar bien los espacios y las tabulaciones del archivo.

Una vez listo para su ejecución utilizamos el comando que viene en la captura para obtener el resultado final:

```
valentingc@valentingcubuntu:~$ ansible-playbook playbook.yml

PLAY [Mostrar por consola] *************************************************************

TASK [Gathering Facts] *************************************************************
ok: [192.168.56.110]
ok: [127.0.0.1]

TASK [Ejecutar el script] *************************************************************
changed: [127.0.0.1]
changed: [192.168.56.110]

PLAY RECAP *************************************************************
127.0.0.1      : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0             ignored=0
192.168.56.110 : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=
0             ignored=0

valentingc@valentingcubuntu:~$ _
```