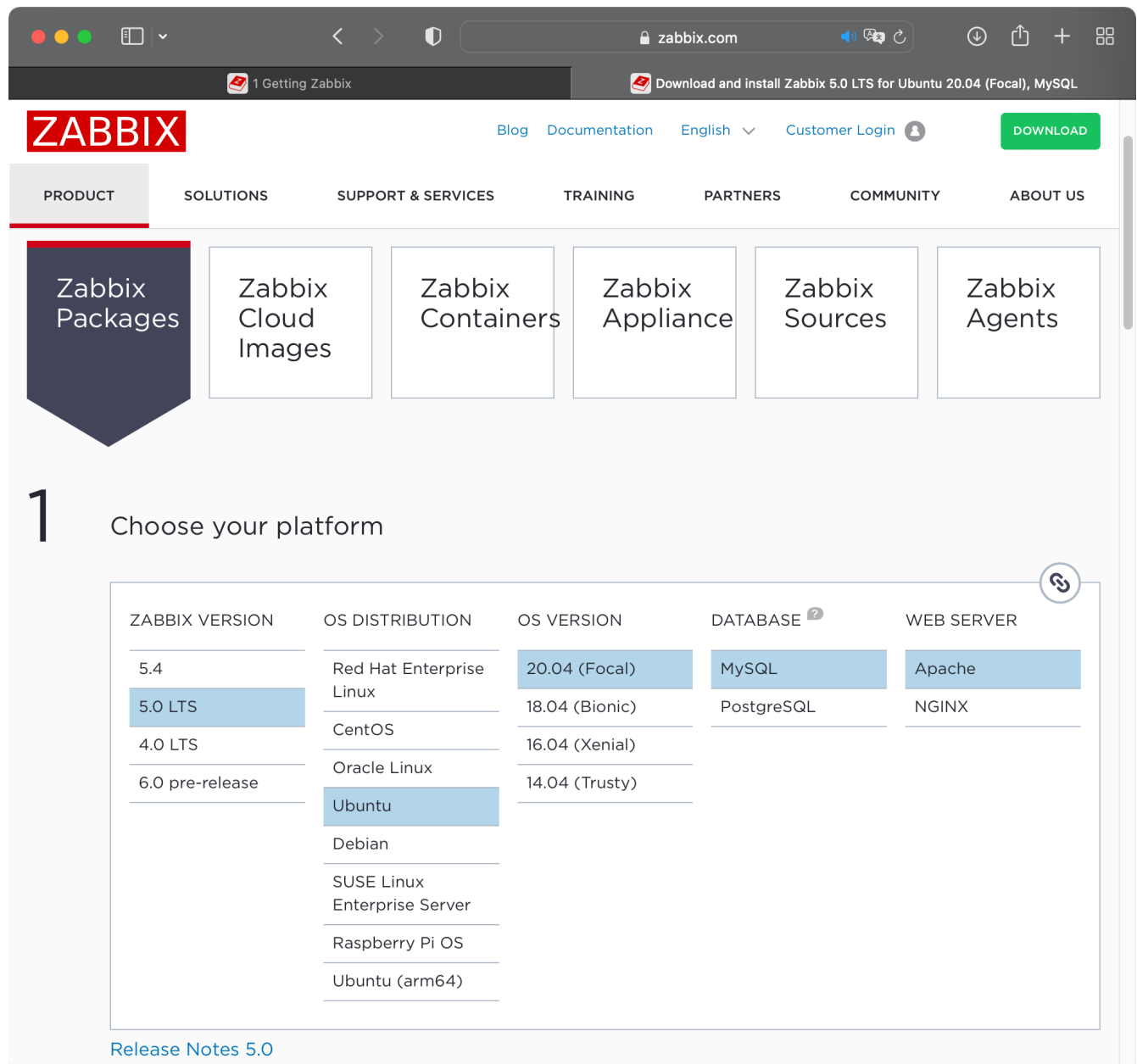


ISE P3 - Monitorización, Automatización y “Profiling”

enero del 2022

Ejercicio 1 - Instalar Zabbix 5.0 en Ubuntu Server 20.04 y configurarlo para que se monitorice a él mismo y para que monitorice a la máquina con CentOS. Se debe de configurar obligatoriamente para que monitorice los servicios SSH y HTTP.

Descargamos el paquete de Zabbix desde la página a la que nos dirige la documentación. Instalamos el repositorio con `sudo dpkg -i zabbix-release_5.0-1+focal_all.deb`. Instalamos Zabbix Server, Frontend y Agente con `sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent`



The screenshot shows the Zabbix website's 'Choose your platform' section. The page features a navigation bar with links to Blog, Documentation, English, Customer Login, and a DOWNLOAD button. Below the navigation bar, there are six product categories: Zabbix Packages, Zabbix Cloud Images, Zabbix Containers, Zabbix Appliance, Zabbix Sources, and Zabbix Agents. The 'Zabbix Packages' category is highlighted. The 'Choose your platform' section includes a table with the following data:

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	DATABASE	WEB SERVER
5.4	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)		
6.0 pre-release	Ubuntu	14.04 (Trusty)		
	Debian			
	SUSE Linux Enterprise Server			
	Raspberry Pi OS			
	Ubuntu (arm64)			

Below the table, there is a link to 'Release Notes 5.0'.

Figure 1: Pre-config de Zabbix

Creamos la base de datos inicial con los comandos que se nos indican en la documentación. Intentamos ejecutar `mysql -uroot -p`, pero nos aparece un error que dice `ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock'`. Investigamos y vemos que es seguramente porque no tenemos instalado mysql server. Después de instalar con `sudo apt install mysql-server` configuramos la bd con

```
sudo mysql_secure_installation
```

```
create database zabbix character set utf8 collate utf8_bin;  
create user zabbix@localhost identified by 'practicas,ise';  
grant all privileges on zabbix.* to zabbix@localhost;  
quit;
```

Descomprimos los datos iniciales de Zabbix con `zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix`.

Modificamos el archivo `/etc/zabbix/zabbix_server.conf` estableciendo `DBPassword=practicas,ise`.

Modificamos el archivo `/etc/zabbix/apache.conf` estableciendo `php_value date.timezone Europe/Madrid` tanto para php5 como php7.

Ejecutamos `sudo systemctl restart zabbix-server zabbix-agent apache2` y `sudo systemctl enable zabbix-server zabbix-agent apache2` para aplicar la configuración y echar a andar los servicios. Este último comando nos solicita múltiples veces la contraseña de nuestro usuario para configurar el servidor y el agente. Intentamos conectarnos por medio de `http://192.168.0.140/zabbix` y lo conseguimos, se nos muestra una pantalla de configuración.

Iniciamos sesión con `Admin/zabbix` y terminamos de configurar.

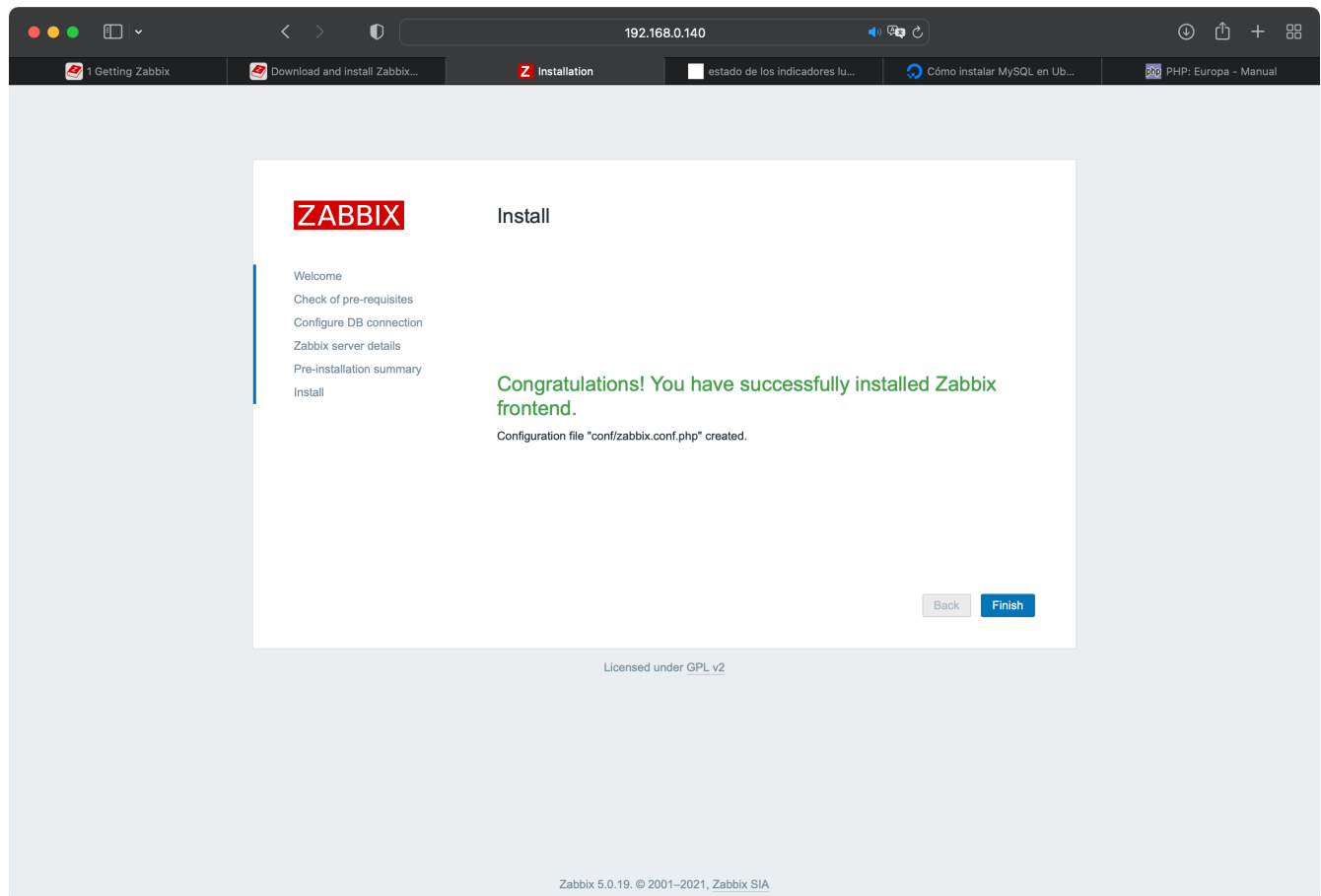
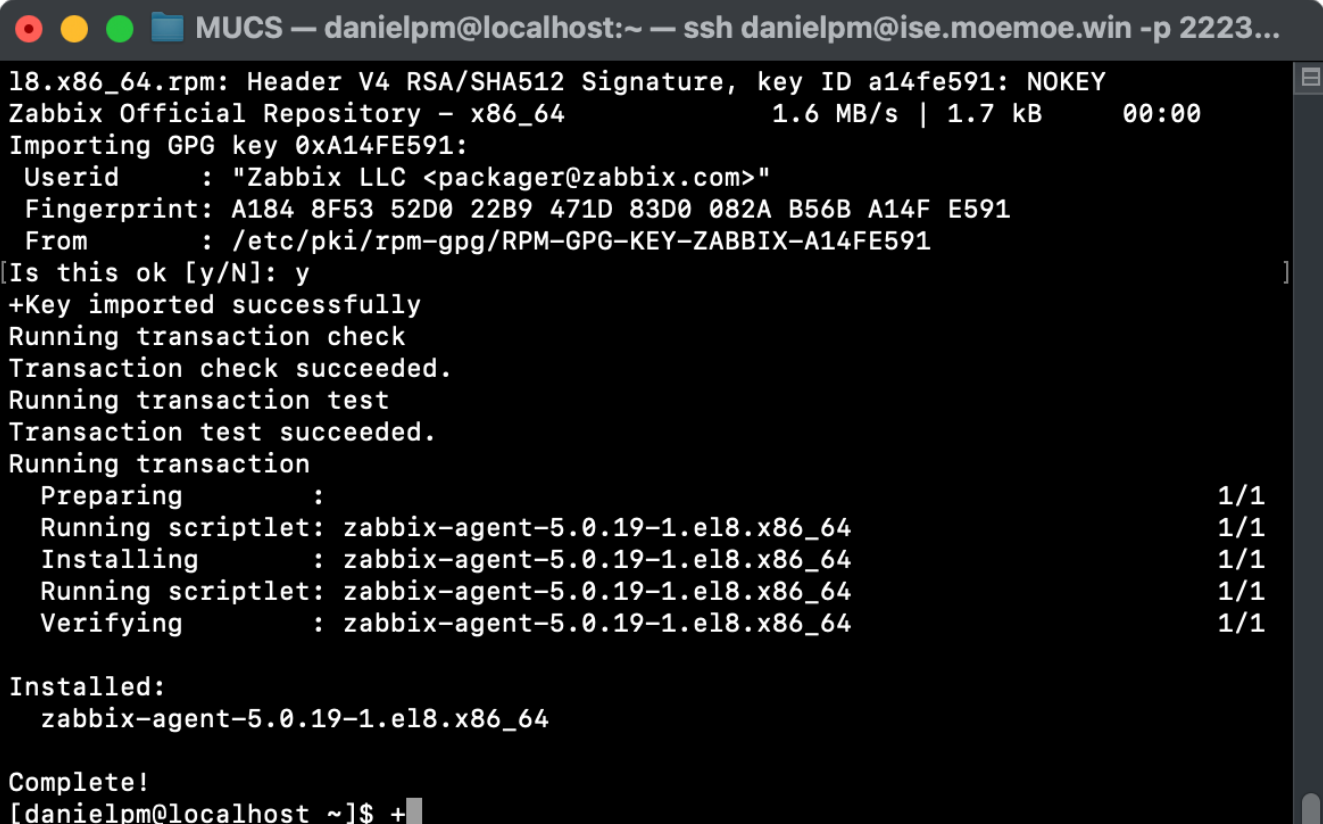


Figure 2: Config-success de Zabbix

Instalación del agente en CentOS

Instalamos el repositorio con `sudo rpm -Uvh https://repo.zabbix.com/zabbix/5.0/rhel/8/x86_64/zabbix-release-5.0-1.el8.x86_64.rpm` y `&& sudo dnf clean all`.

Instalamos el agente con `sudo dnf install zabbix-agent`.



```
18.x86_64.rpm: Header V4 RSA/SHA512 Signature, key ID a14fe591: NOKEY
Zabbix Official Repository - x86_64 1.6 MB/s | 1.7 kB 00:00
Importing GPG key 0xA14FE591:
  Userid      : "Zabbix LLC <packager@zabbix.com>"
  Fingerprint: A184 8F53 52D0 22B9 471D 83D0 082A B56B A14F E591
  From        : /etc/pki/rpm-gpg/RPM-GPG-KEY-ZABBIX-A14FE591
[Is this ok [y/N]: y
+Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      : zabbix-agent-5.0.19-1.el8.x86_64 1/1
  Running scriptlet: zabbix-agent-5.0.19-1.el8.x86_64 1/1
  Installing      : zabbix-agent-5.0.19-1.el8.x86_64 1/1
  Running scriptlet: zabbix-agent-5.0.19-1.el8.x86_64 1/1
  Verifying       : zabbix-agent-5.0.19-1.el8.x86_64 1/1

Installed:
  zabbix-agent-5.0.19-1.el8.x86_64

Complete!
[danielpm@localhost ~]$
```

Figure 3: Instalación agente

Encontramos instrucciones para configurar el agente en <https://davidshomelab.com/install-zabbix-agent-to-monitor-windows/>.

Editamos el archivo `/etc/zabbix/zabbix_agentd.conf` indicando la IP del servidor de Zabbix y el hostname.

Añadimos el puerto de Zabbix al firewall mediante `sudo firewall-cmd --add-port=10050/tcp --permanent --zone=public`. Aplicamos la regla con `sudo firewall-cmd --reload`.

Inicializamos el servicio con `sudo systemctl enable zabbix-agent && sudo systemctl restart zabbix-agent`.

Comprobamos el estado con `sudo systemctl status zabbix-agent`.

Conexión del agente Zabbix al servidor Zabbix

Nos dirigimos desde el front-end de zabbix a configuration -> hosts -> create host. Creamos especificando como IP 192.168.0.141 y nombre CentOS 8.

Añadimos una template al agente especificando en Hosts -> CentOS 8 -> Templates, **Template OS Linux by Zabbix agent**. Esperamos unos segundos y tras refrescar la página comprobamos que el servidor tiene conexión con el agente ya que en la fila de CentOS 8 está marcado en verde el botón de ZBX.

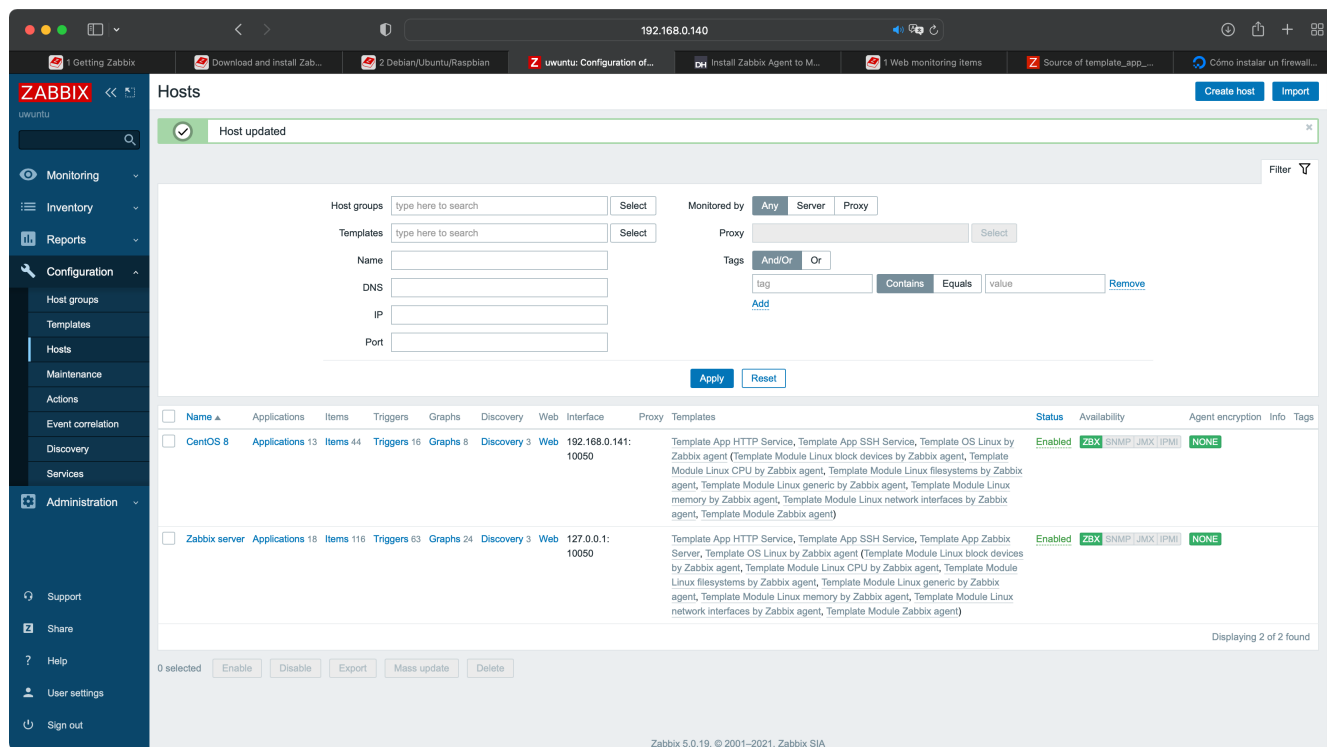


Figure 4: Añadir el agente al servidor con un template asignado

Monitorización de SSH y HTTP

Desde configuration -> hosts clicamos en cada uno de los hosts y le asignamos las plantillas **Template App HTTP Service** y **Template App SSH Service**.

Desde el dashboard podemos ver cómo CentOS 8 nos indica que el servicio HTTP está caído. Esto es esperable, y nos sirve para comprobar que funciona correctamente, ya que el servicio HTTP no está habilitado ni corriendo en este momento en la máquina en la que tenemos instalado CentOS 8.

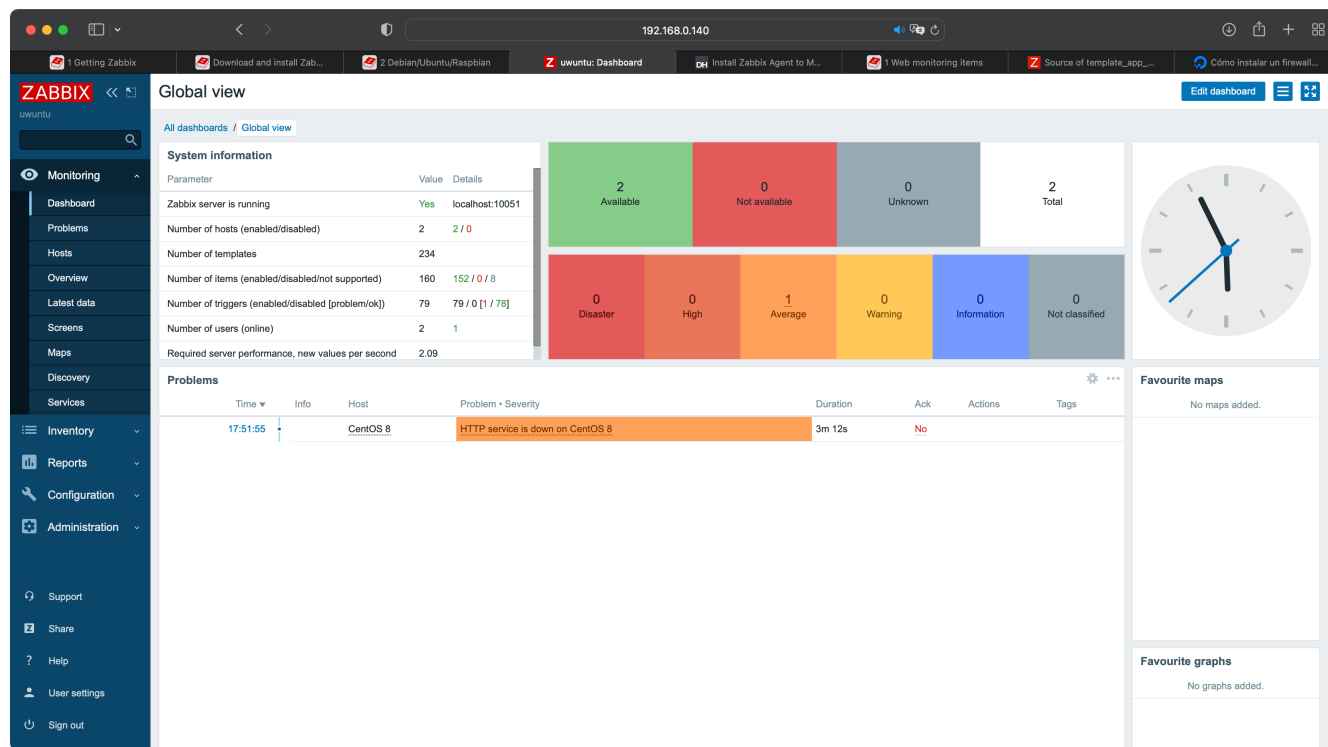


Figure 5: HTTP caído en el agente porque no está instalado

Ejercicio 2 - Instalar y configurar Ansible para poder hacer un ping a las máquinas virtuales de los servidores y ejecutar un comando básico. Posibilidad de hacer acciones más complejas mediante playbooks escritos con YAML. Incluir capturas de pantalla del proceso.

Instalando Ansible

Src: UNIX and Linux System Administration Handbook.

Podemos instalar Ansible mediante apt, ejecutando `sudo apt install ansible`.

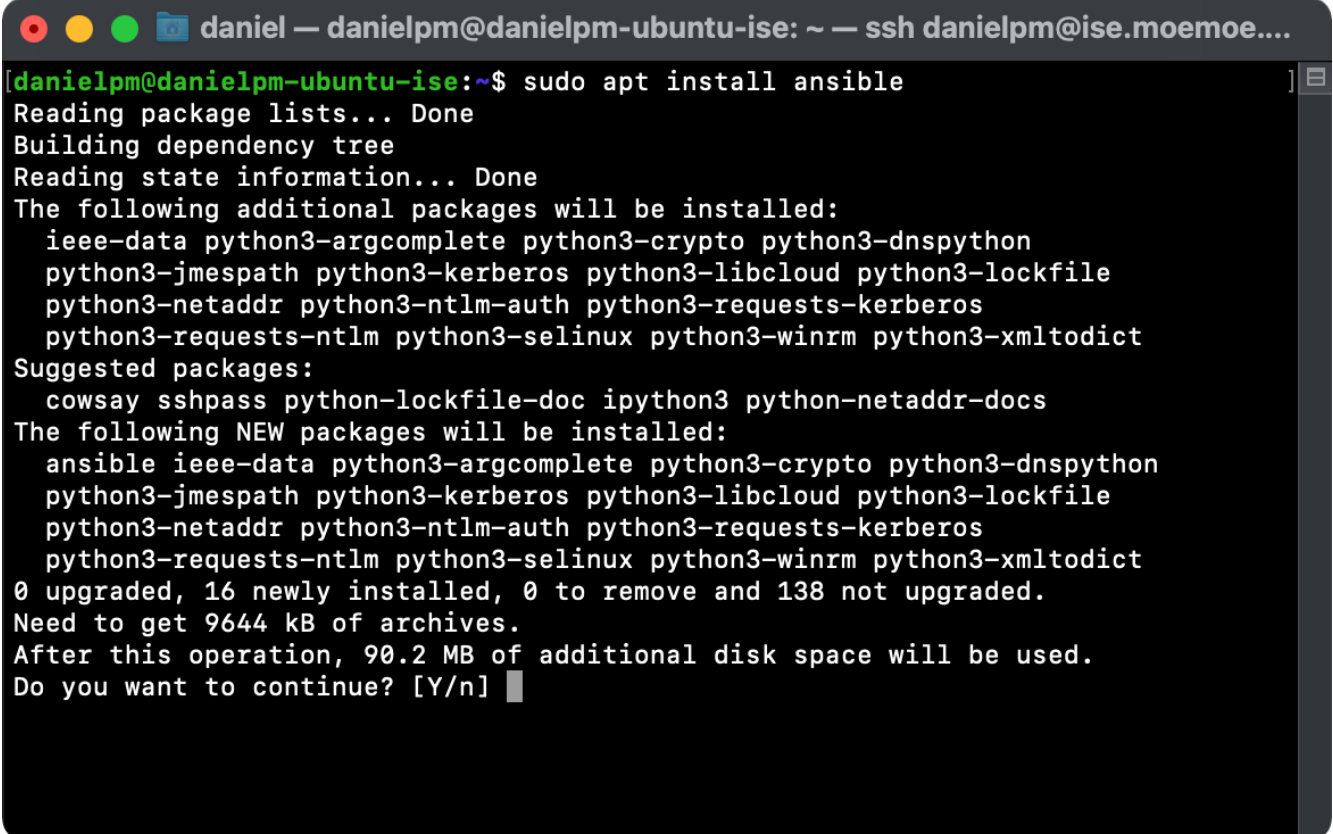
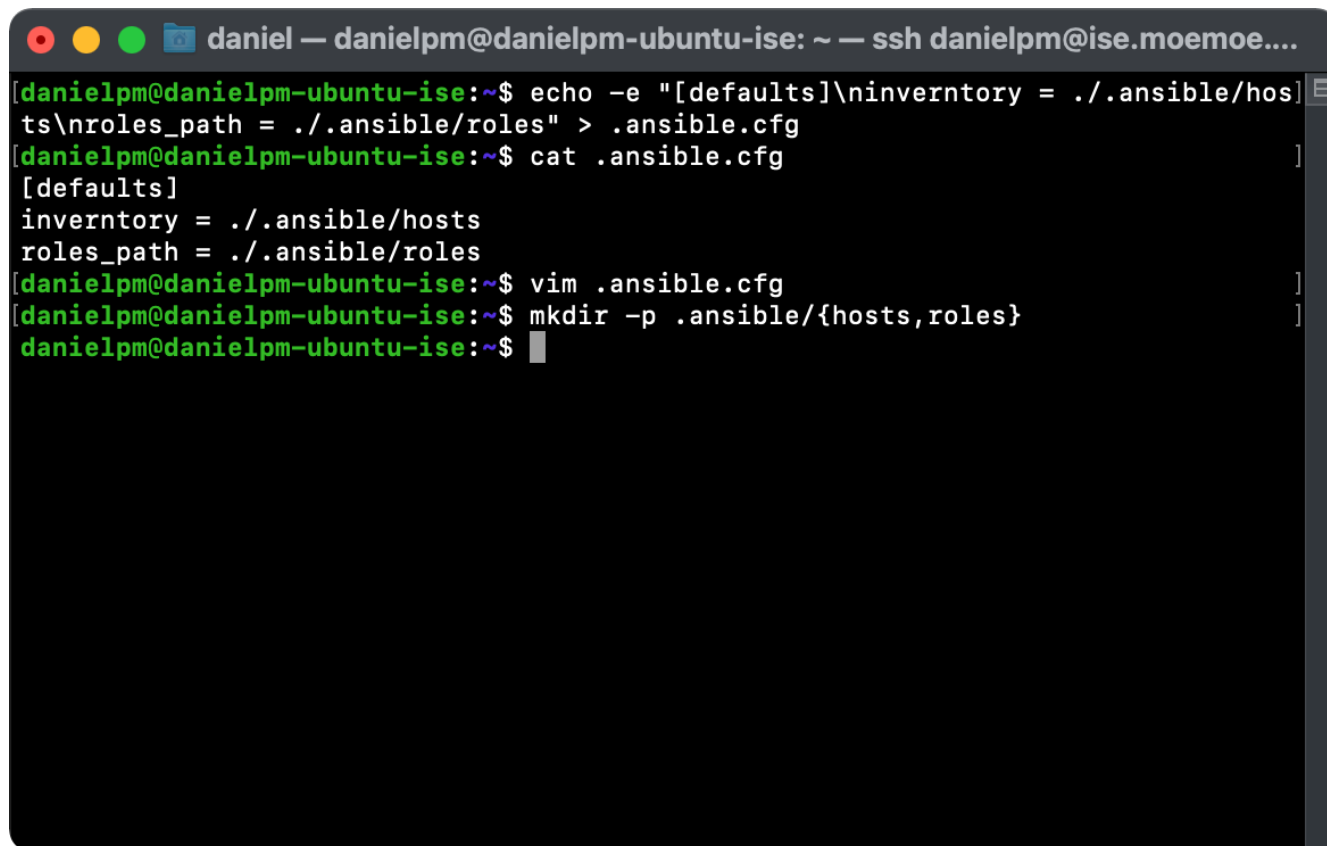
A terminal window titled 'daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....'. The prompt is '[danielpm@danielpm-ubuntu-ise:~\$]'. The user enters 'sudo apt install ansible'. The output shows the package lists being read, the dependency tree being built, and state information being read. It lists additional packages to 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, and python3-xlrd. It also lists suggested packages: cowsay, sshpass, python-lockfile-doc, ipython3, and python-netaddr-docs. The new packages to be installed are listed: 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, and python3-xlrd. The summary shows 0 upgraded, 16 newly installed, 0 to remove, and 138 not upgraded. The total size of the archives is 9644 kB. The additional disk space required is 90.2 MB. The prompt asks 'Do you want to continue? [Y/n]' with a cursor.

Figure 1: Instalación de Ansible en Ubuntu Server 20.04

Configurando Ansible

Para configurar ansible sin tener que modificar el archivo de configuración por defecto del sistema creamos el archivo `~/.ansible.cfg` donde indicaremos que el inventario y roles los tendremos en la carpeta `~/.ansible`.

A terminal window with a dark background and light-colored text. The window title is "daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....". The terminal shows the following commands and output:

```
[danielpm@danielpm-ubuntu-ise:~$ echo -e "[defaults]\ninverntory = ~/.ansible/hos]
ts\nroles_path = ~/.ansible/roles" > .ansible.cfg
[danielpm@danielpm-ubuntu-ise:~$ cat .ansible.cfg
[defaults]
inverntory = ~/.ansible/hosts
roles_path = ~/.ansible/roles
[danielpm@danielpm-ubuntu-ise:~$ vim .ansible.cfg
[danielpm@danielpm-ubuntu-ise:~$ mkdir -p .ansible/{hosts,roles}
[danielpm@danielpm-ubuntu-ise:~$
```

Figure 2: Configuración de Ansible para nuestro usuario

Seguidamente comprobamos que está tanto instalado python3 (en CentOS no lo está, en Ubuntu sí) como el usuario ansible creado tanto en CentOS como en Ubuntu con permisos para utilizar sudo (creamos el usuario con `sudo useradd ansible --gid wheel && sudo passwd ansible` en CentOS y con `sudo useradd ansible --gid sudo && sudo passwd ansible` en Ubuntu).

Dentro del archivo `~/.ansible/hosts/static` definimos los hosts estáticos, en nuestro caso CentOS y Ubuntu. El fichero queda así:

```
192.168.0.141 ansible_user=ansible
localhost ansible_user=ansible
```

Hacemos una prueba para comprobar si hemos configurado todo bien. Nos sale el siguiente error:

Intuímos que por el mensaje de error que requiere login sin contraseña. Generamos una identidad y la copiamos a CentOS.

Volvemos a probar y esta vez conseguimos conectarnos a la máquina.

Intentamos también utilizar sudo, para comprobar que funciona correctamente. Funciona correctamente.

Hacemos lo mismo en localhost. Vemos que useradd no ha creado una carpeta home para Ansible. La creamos y comprobamos igualmente que funciona correctamente.


```
daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....
[danielpm@danielpm-ubuntu-ise:~$ sudo apt install python3
[sudo] password for danielpm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.8.2-0ubuntu2).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 138 not upgraded.
[danielpm@danielpm-ubuntu-ise:~$ sudo useradd ansible --gid sudo
[danielpm@danielpm-ubuntu-ise:~$ sudo passwd ansible
New password:
Retype new password:
passwd: password updated successfully
[danielpm@danielpm-ubuntu-ise:~$ ansible 192.168.0.141 -a whoami
192.168.0.141 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ansible@192.168.0.141: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).",
  "unreachable": true
}
[danielpm@danielpm-ubuntu-ise:~$
```

Figure 3: Error *host unreachable* en Ansible al intentar ejecutar `whoami`

Creamos ahora un grupo en `~/.ansible/hosts/hosts` para poder referirnos a ambos servidores con un único nombre.

```
[linuxservers]
192.168.0.141
localhost
```

Comprobamos que funciona en el siguiente apartado.

Ping a las máquinas virtuales

Realizamos un ping ejecutando `ansible linuxservers -m ping`. Comprobamos que funciona todo correctamente.

Configuración de un rol sencillo

Src: https://docs.ansible.com/ansible/latest/user_guide/playbooks_reuse_roles.html

Creamos bajo `~/.ansible/roles/get_all_users/tasks` el archivo `main.yml`, donde indicamos lo siguiente:

```
- name: Get connected users
  shell: "who"

- name: Get current list of usernames
  shell: "cut -d: -f1 /etc/passwd"
```

Probamos que funciona correctamente el rol ejecutando `ansible linuxservers -m include_role -a name=get_all_users`

Src: <https://stackoverflow.com/questions/38350674/ansible-can-i-execute-role-from-command-line>

```
daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe...

[danielpm@danielpm-ubuntu-ise:~$ ssh-copy-id ansible@192.168.0.141
/usr/bin/ssh-copy-id: ERROR: No identities found
[danielpm@danielpm-ubuntu-ise:~$ ssh
ssh                ssh-argv0          ssh-import-id-gh  ssh-keyscan
ssh-add            ssh-copy-id          ssh-import-id-lp  sshd
ssh-agent          ssh-import-id        ssh-keygen
[danielpm@danielpm-ubuntu-ise:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/danielpm/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/danielpm/.ssh/id_rsa
Your public key has been saved in /home/danielpm/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:1tcueId8MB49N9JlZY2Q8cIBdr4gz9VyLViXbTvZmsA danielpm@danielpm-ubuntu-ise
The key's randomart image is:
+---[RSA 3072]-----+
|      o.=+..B|
|      . +.*.==|
|      . ..B =.B|
|      = oEB.*o|
|      S + *o+=o|
|      .   = *+o.|
|      . * +   |
|      . +     |
+-----[SHA256]-----+
[danielpm@danielpm-ubuntu-ise:~$ ssh-copy-id ansible@192.168.0.141
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/danielpm/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
ansible@192.168.0.141's password:
Permission denied, please try again.
ansible@192.168.0.141's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'ansible@192.168.0.141'"
and check to make sure that only the key(s) you wanted were added.

danielpm@danielpm-ubuntu-ise:~$
```

Figure 4: Creación y copia de nuestra identidad hacia el host ejecutando CentOS

```
daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....
[danielpm@danielpm-ubuntu-ise:~$ ansible 192.168.0.141 -a whoami
192.168.0.141 | CHANGED | rc=0 >>
ansible
[danielpm@danielpm-ubuntu-ise:~$ ansible 192.168.0.141 -a whoami --become --ask-b]
ecome-pass
[BECOME password:
192.168.0.141 | CHANGED | rc=0 >>
root
danielpm@danielpm-ubuntu-ise:~$
```

Figure 5: Ejecución de whoami en el host ejecutando CentOS mediante Ansible, con y sin privilegios de ejecución

```
daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....
[danielpm@danielpm-ubuntu-ise:~$ vim .ansible/hosts/hosts
[danielpm@danielpm-ubuntu-ise:~$ cat .ansible/hosts/hosts
[linuxservers]
192.168.0.141
localhost
[danielpm@danielpm-ubuntu-ise:~$ ansible 'linuxservers' -a whoami
192.168.0.141 | CHANGED | rc=0 >>
ansible
localhost | CHANGED | rc=0 >>
ansible
[danielpm@danielpm-ubuntu-ise:~$ ansible 'linuxservers' -m ping
localhost | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.0.141 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
danielpm@danielpm-ubuntu-ise:~$
```

Figure 6: whoami y tarea ping al grupo linuxservers en Ansible

```
daniel — danielpm@danielpm-ubuntu-ise: ~ — ssh danielpm@ise.moemoe....

[danielpm@danielpm-ubuntu-ise:~$ ansible linuxservers -m get_all_users
192.168.0.141 | FAILED! => {
  "msg": "The module get_all_users was not found in configured module paths"
}
localhost | FAILED! => {
  "msg": "The module get_all_users was not found in configured module paths"
}

[danielpm@danielpm-ubuntu-ise:~$ ansible linuxservers -m include_role -a name=get_all_users
192.168.0.141 | SUCCESS => {
  "changed": false,
  "include_args": {
    "name": "get_all_users"
  }
}
localhost | SUCCESS => {
  "changed": false,
  "include_args": {
    "name": "get_all_users"
  }
}

192.168.0.141 | CHANGED | rc=0 >>
ansible pts/0      2021-12-28 05:58 (192.168.0.140)
localhost | CHANGED | rc=0 >>
danielpm pts/0     2021-12-28 08:24 (192.168.0.20)
ansible pts/3      2021-12-28 10:58 (127.0.0.1)
localhost | CHANGED | rc=0 >>
root
daemon
bin
sys
sync
games
man
lp
mail
news
```

Figure 7: Ejecución de un *role* sobre el grupo `linuxservers` en Ansible