



UNIVERSIDAD
DE GRANADA

Cloud Computing: Servicios y Aplicaciones

E.T.S.I. Informática y Telecomunicación

Máster en Ingeniería Informática

Universidad de Granada

Curso académico 2018-2019

Práctica 1

DISEÑO Y DESPLIEGUE DE APLICACIONES SOBRE PLATAFORMAS
CLOUD

Correa Fernández, Gema 75572158-T

gecorrea@correo.ugr.es

5 de mayo de 2019

Índice

| | |
|------------------------------------------------------|-----------|
| 1. Descripción del problema | 1 |
| 2. Diseño | 1 |
| 3. Despliegue | 3 |
| 3.1. Instalación | 3 |
| 3.1.1. Servicio ownCloud | 4 |
| 3.1.2. Servicio MySQL/MariaDB | 7 |
| 3.1.3. Servicio LDAP | 11 |
| 3.2. Configuración de LDAP en ownCloud | 14 |
| 3.3. Comprobar conexión ownCloud y MariaDB | 19 |
| 4. Manual de usuario | 20 |
| 5. Mejoras | 22 |
| Referencias | 23 |

Práctica 1: Diseño y despliegue de aplicaciones sobre plataformas cloud

1. Descripción del problema

El objetivo de esta primera práctica es poner en uso los conocimientos adquiridos en relación con el uso de plataformas de cloud computing a nivel de IaaS y SaaS. Para ello, se deberá realizar el diseño y posterior despliegue de una aplicación para gestión de archivos. La aplicación que se desea desplegar precisa de los siguientes servicios:

- Autenticación basada en LDAP.
- Alojamiento, sincronización y compartición de archivos.
- Base de datos.

Para ello deberemos hablar de la virtualización a nivel de sistema operativo, la cuál es un tipo de virtualización en donde la capa de virtualización se ejecuta como una aplicación en el sistema operativo. En este caso, el kernel del SO se ejecuta sobre el nodo del hardware con varias máquinas virtuales invitadas aisladas [1].

En este trabajo se pondrá en práctica la creación de varias máquinas virtuales para la realización de nuestra aplicación. En concreto, se va hacer uso de una máquina virtual y dentro de ella un contenedor docker con el servicio determinado, es decir, se crearán tres máquinas virtuales (arquitectura de microservicios):

- Una máquina virtual y dentro un contenedor que contenga la identificación de usuarios mediante LDAP.
- Una máquina virtual y dentro un contenedor que contenga la web de ownCloud.
- Una máquina virtual y dentro un contenedor que contenga la base de datos MySQL/MariaDB.

Estas máquinas virtuales se desplegarán haciendo uso la plataforma cloud de Microsoft Azure. A continuación, se detalla el procedimiento seguido para cumplimentar los objetivos de dicha práctica.

2. Diseño

Para el despliegue de la aplicación en la nube, se ha escogido como plataforma cloud, **Microsoft Azure**, debido a que es la única en la que se poseía crédito para el despliegue de la aplicación, además de que ya se han desarrollado prácticas de otras asignaturas en ella, por lo que no parto de un conocimiento desde cero. Además, dicha plataforma es una de las más usadas en el mercado, y es bueno disponer de dicho conocimiento.

El software elegido para el almacenamiento, sincronización y compartición de archivos ha sido **ownCloud**, por ser de software libre y permitir almacenar archivos con una interfaz ya conocida. Esta última justificación, ha sido la que ha supuesto la elección de dicho software. Además, se pensó en hacer uso de *Nextcloud*, pero al ser un un software

relativamente más nuevo que ownCloud, se ha leído en tutoriales que la redacción en inglés de los manuales online no es muy extensa hasta la fecha. Asimismo, las actualizaciones no tienen lugar de una manera tan sencilla como en ownCloud [2].

Tras saber que es posible conectar una base de datos a ownCloud, se ha escogido a usar **MySQL/MariaDB**, esta elección ha sido escogida por haber trabajado con MySQL en la asignatura de Cloud Computing del primer cuatrimestre. En nuestro caso, se ha escogido la imagen de MariaDB que está basada en MySQL. En principio, tenía escogido hacer uso de MySQL, pero en clase se comentó que daría problemas su conexión, por lo que he optado por escoger una variante similar.

Para la autenticación de usuarios, se ha escogido **LDAP**, debido a que se comentó en clase y ya se había trabajado con ella. De todas maneras, se buscaron alternativas y se encontró *Active Directory* que también es un sistema basado en bases de datos que proporciona autenticación, directorio, política y otros servicios, pero en un entorno Windows, y nuestras máquinas virtuales hacen uso de Ubuntu [3].

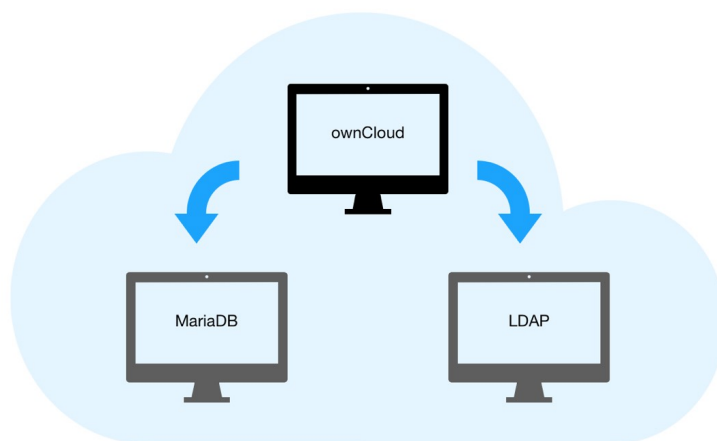


Figura 1: Diseño de la aplicación

Se ha optado por crear la base de datos desde cero debido a que el servicio de MariaDB, no permite acceso desde el exterior. En nuestra aplicación, será el servicio de ownCloud quien se encargue de comunicarse con la base de datos y con el servicio de autenticación de usuarios.

Por otro lado, el diseño del despliegue de la aplicación partió de tres hipótesis:

1. Realizar una máquina virtual, y dentro disponer de un *docker-compose*, con los tres servicios.
2. Realizar un contenedor por servicio.
3. **Realizar un máquina virtual por servicio, y dentro el contenedor con dicho servicio.**

De las tres posibles opciones que acabo de comentar, se escogió la última, debido a la facilidad de disponer de tres máquinas virtuales individuales, las cuales estarán co-

nectadas entre sí, ya que éste permite una mejor **escalabilidad**. Además, en clase se comentó esta posibilidad. Por otro lado, Azure nos permite realizar fácilmente un escalado automático de máquinas virtuales [4]. Ya que Microsoft Azure es la forma más sencilla de realizar escalabilidad de los recursos de una máquina, es tan sencillo como hacer tres *clicks* y aumentar la CPU y memoria RAM. Lo que permite que podamos crear multitud de contenedores, dentro de cada una de ellas. Es por eso que para garantizar la escalabilidad de la aplicación, se debe garantizar el tamaño de las máquinas virtuales. Y al montarse en una plataforma cloud es posible, de acuerdo siempre al crédito disponible. Por eso se han creado tres máquinas virtuales independientes.

3. Despliegue

En este apartado se describirá como se despliegan los servicios que componen la aplicación con el fin de ponerla en marcha para su posterior uso. Antes de comenzar con los pasos, debe ser importante disponer de una cuenta en Microsoft Azure y tener instalado Azure CLI, puesto que el despliegue y la creación de las distintas instancias, se ha hecho haciendo uso de dicha plataforma cloud.

El primer paso será instalar y lanzar los tres servicios, para luego su posterior correcta configuración y puesta en marcha.

Nota: Como el código está disponible en Github, no se van a poner todas las sentencias usadas.

3.1. Instalación

El primer paso para poner en marcha nuestra aplicación, es disponer de los módulos necesarios. Para ello, lo primero que se debe realizar es lanzar el script `recursos.sh` con la sentencia `sh recursos.sh`. En dicho script, disponemos de la creación del **grupo de recursos** y la **red virtual** de Azure donde se van a lanzar las tres máquinas:

The screenshot shows the Azure portal interface for a virtual network named 'myVnet'. The left sidebar contains a navigation menu with options like 'Agregar', 'Filtrar por nombre...', 'NOMBRE', and 'myVnet'. The main area displays the 'Información general' (General information) tab for 'myVnet'. It shows details such as the resource group 'myResourceGroup-eastus', location 'Este de EE. UU.', subscription 'Hito 5: uso de Vagrant GEMA CORREA FERNANDEZ', and the address space '10.0.0.0/16'. Below this, the 'Dispositivos conectados' (Connected devices) section shows a table with three connected devices: 'MV1-owncloudVMNic', 'MV3-ldapVMNic', and 'MV2-databaseVMNic', all of type 'Interfaz de red' (Network interface) and connected to the 'mySubnet'.

| DISPOSITIVO | TIPO | DIRECCIÓN IP | SUBRED |
|-------------------|-----------------|--------------|----------|
| MV1-owncloudVMNic | Interfaz de red | 10.0.1.4 | mySubnet |
| MV3-ldapVMNic | Interfaz de red | 10.0.1.5 | mySubnet |
| MV2-databaseVMNic | Interfaz de red | 10.0.1.6 | mySubnet |

Figura 2: Red virtual una vez creadas las tres máquinas

```
#!/bin/bash
```

```
# Creacion del grupo de recursos con localizacion en el este de EE.UU.
echo " — Creacion del grupo de recursos — "
az group create --name myResourceGroup-eastus --location eastus

# Creacion de una red virtual y una subred
echo " — Creacion de la red virtual — "
az network vnet create --resource-group myResourceGroup-eastus ...
```

3.1.1. Servicio ownCloud

Una vez que disponemos de nuestro grupo de recursos, pasamos a la creación de la primera máquina virtual, la cual contiene el servicio destinado a **ownCloud** para el almacenamiento, sincronización y compartición de archivos. Para ello, se ha creado un script `maquina1-owncloud.sh` (`sh maquina1-owncloud.sh`), el cual nos crea una máquina virtual, con un tamaño específico y los requerimientos necesarios para poder acceder a dicha máquina. Para dicho servicio, solo es necesario disponer del puerto 80 abierto.

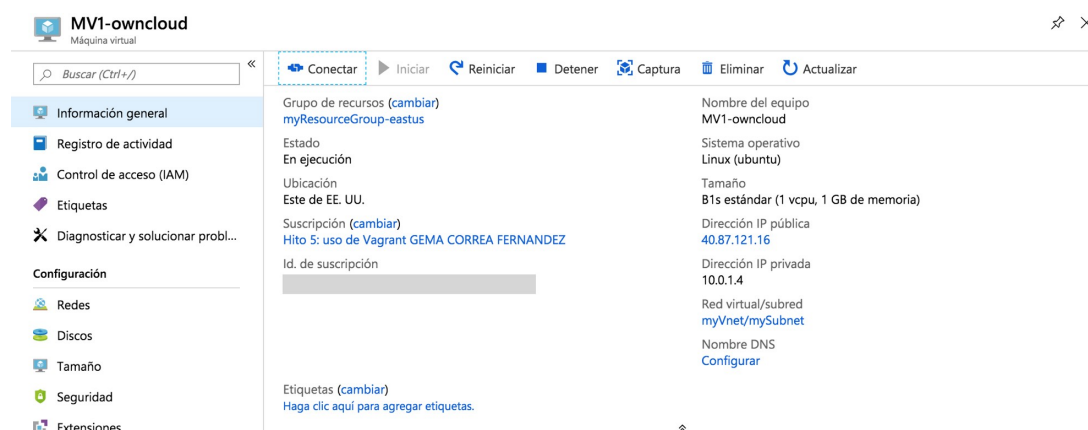


Figura 3: Máquina virtual creada

```
#!/bin/bash

# Creacion de la maquina virtual 1 para Owncloud
# Obtenemos la IP
echo " — Creacion de la maquina virtual para ownCloud — "
IP=$(az vm create ... | jq -r '.publicIpAddress')

# Cambiamos la dimension a la maquina virtual
echo " — Cambiando la dimension a la maquina virtual — "
az vm resize --resource-group myResourceGroup-eastus ...

# Abrimos el puerto 80
echo " — Abrimos el puerto 80 — "
az vm open-port --port 80 --resource-group myResourceGroup-eastus ...

# Una vez creada, mostramos su nombre y su direccion IP
```

```
echo " ——— Datos de la maquina virtual creada para Owncloud ——— "
echo -name: MV1-owncloud
echo -ip: $IP
```

Podemos encontrar dicha máquina en Azure, con el nombre de *MV1-owncloud*. Una vez que hayamos lanzado el script, nos aparecerá en la terminal, una ventana como la siguiente. Así que deberemos escribir en la consola `ssh gemazure@<IP>` para así poder acceder a ella, se deberá poner la IP que se haya asignado a la máquina, visible desde Azure o desde el terminal. Una vez creada la máquina virtual y con acceso a ella, instalaremos el contenedor correspondiente a la última imagen de ownCloud:

```
# Conectarnos a la maquina virtual
ssh gemazure@$IP

# Instalar docker
sudo apt install docker.io

# Descargar owncloud y ponerlo en funcionamiento
sudo docker pull owncloud:latest
sudo docker run -d -p 80:80 owncloud:latest
```

```
----- Datos de la máquina virtual creada para Owncloud -----
-name: MV1-owncloud
-ip: 40.87.121.16
MacBook-Pro-de-Gema:Práctica 1 gema$ ssh gemazure@40.87.121.16
The authenticity of host '40.87.121.16 (40.87.121.16)' can't be established.
ECDSA key fingerprint is SHA256:ly0pdqoNkXD8h5uuyQ0/OU6vVuYw6t2PevDZyIK1UZs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.87.121.16' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-1014-azure x86_64)

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

System information as of Sat May  4 12:22:29 UTC 2019

System load:  0.75           Processes:           117
Usage of /:   4.1% of 28.9GB Users logged in:       0
Memory usage: 25%           IP address for eth0: 10.0.1.4
Swap usage:   0%

2 packages can be updated.
2 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Figura 4: Accediendo a la máquina virtual por SSH

```
gemazure@MV1-owncloud:~$ sudo docker pull owncloud:latest
latest: Pulling from library/owncloud
177e7ef0df69: Pull complete
9bf89f2eda24: Pull complete
350207dcf1b7: Pull complete
a8a33d96b4e7: Pull complete
c0421d5b63d6: Pull complete
f76e300f6e72: Pull complete
af9ff1b9ce5b: Pull complete
d9f072d61771: Pull complete
a6c512d0c2db: Pull complete
5a99458af5f8: Pull complete
8f2842d661a0: Pull complete
3c71c5361f06: Pull complete
baeacbad0a0c: Pull complete
e60049bf081a: Pull complete
0619078e32d3: Pull complete
a8e482ee2313: Pull complete
174d1b06857d: Pull complete
4a86c437f077: Pull complete
5e9ed4c3df2d: Pull complete
8a1479477c8e: Pull complete
8ab262044e9e: Pull complete
Digest: sha256:173811cb4c40505401595a45c39a802b89fb476885b3f6e8fe327aae08d20fe8
Status: Downloaded newer image for owncloud:latest
gemazure@MV1-owncloud:~$ sudo docker run -d -p 80:80 owncloud:latest
a7257d67357a4d70c2184b89aae1d63b1d9954ccae7402ad403bd9e87183978f
```

Figura 5: Descargando el contenedor de ownCloud

Y ya solo, hace falta poner la dirección IP en el navegador (<http://40.87.121.16/>) y el usuario (*admin*) y la contraseña (*admin*):

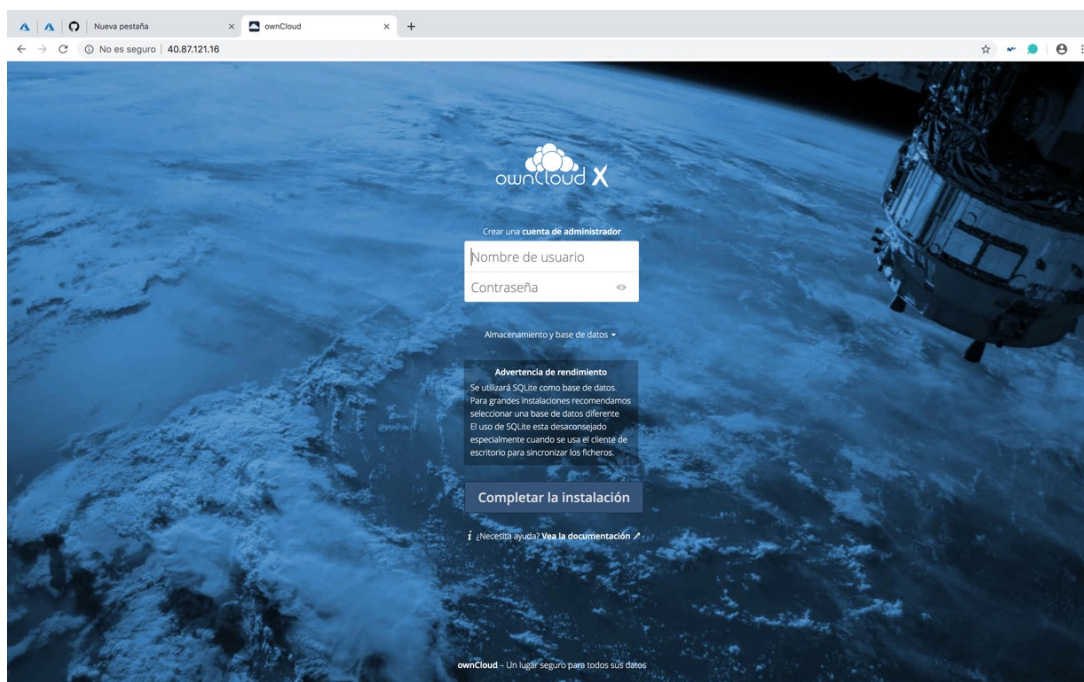


Figura 6: Página inicial de ownCloud

3.1.2. Servicio MySQL/MariaDB

Considerando que tenemos instalado correctamente el servicio ownCloud y funcionando, pasamos a la creación del servicio destinado para la base de datos, en este caso se ha hecho uso de MySQL/MariaDB. Para ello, se ha creado un script `maquina2-mariadb.sh` (`sh maquina2-mariadb.sh`), el cual nos crea una máquina virtual, con un tamaño específico y los requerimientos necesarios para poder acceder a dicha máquina. Para dicho servicio, solo es necesario disponer del puerto 3306 abierto, el correspondiente al puerto de MySQL. Para la creación de dicho servicio se ha seguido el tutorial [5].

```
#!/bin/bash

# Creacion de la maquina virtual 2 para BD
echo " ————— Creacion de la maquina virtual para BD ————— "
IP2=$(az vm create ... | jq -r '.publicIpAddress')

# Abrimos el puerto 3306
echo " ————— Abrimos el puerto 3306 ————— "
az vm open-port --port 3306 ...

# Una vez creada, mostramos su nombre y su direccion IP
echo " ————— Datos de la maquina virtual creada para BD ————— "
echo -name: MV2-database
echo -ip: $IP2
```

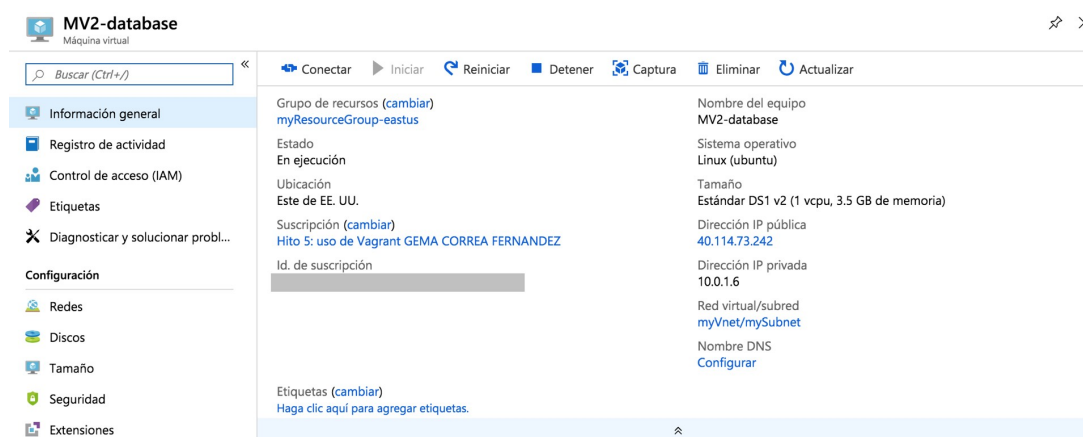


Figura 7: Máquina virtual creada

Podremos encontrar la máquina creada en Azure, con el nombre de MV2-database. Una vez creada, pasamos a la instalación del contenedor y docker destinado para el servicio, después de acceder a ella por SSH (`ssh gemazure@<IP2>`) (figura 8).

Cuando hemos entrado en la máquina virtual, es necesario instalar el contenedor destinado para el servicio. Como se comentó en clase, es posible que MySQL/MariaDB dé problemas con el servicio ownCloud, además se debe abrir el servicio para que permita acceso desde el exterior, es por eso que se ha considerado realizar una imagen de Ma-

riaDB de acuerdo a nuestras especificaciones, en vez de buscar la que pueda funcionar en Docker Hub. A continuación, podemos ver el `Dockerfile` creado para ello.

```
----- Datos de la máquina virtual creada para BD -----
-name: MV2-database
-ip: 40.114.73.242
MacBook-Pro-de-Gema:Práctica 1 gema$ ssh gemazure@40.114.73.242
The authenticity of host '40.114.73.242 (40.114.73.242)' can't be established.
ECDSA key fingerprint is SHA256:leaQmkiGb1UYPIaM44cXCQJRPkp1Tf44Jg4PkjEDy9A.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.114.73.242' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-1014-azure x86_64)

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

System information as of Sat May  4 12:33:38 UTC 2019

System load:  0.23           Processes:            116
Usage of /:   4.1% of 28.90GB Users logged in:      0
Memory usage: 7%           IP address for eth0: 10.0.1.6
Swap usage:   0%

2 packages can be updated.
2 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

gemazure@MV2-database:~$
```

Figura 8: Accediendo a la máquina virtual por SSH

```
# Se utiliza la imagen de Ubuntu Trusty
FROM ubuntu:trusty

RUN apt-get update
RUN DEBIAN_FRONTEND=noninteractive apt-get install -y mariadb-server
# pwgen: generador de password fuertes para sistemas linux
RUN DEBIAN_FRONTEND=noninteractive apt-get install -y pwgen

# Eliminar si esta instalado MySQL/MariaDB
RUN rm -rf /var/lib/mysql/*

# Cambiar: bind address to 0.0.0.0
RUN sed -i -r 's/bind-address.*$/bind-address = 0.0.0.0/'
    /etc/mysql/my.cnf

ADD crear_usuario.sh /crear_usuario.sh
ADD ejecutar.sh /ejecutar.sh

# Dar permisos de ejecucion, lectura y escritura
RUN chmod 775 /*.sh

# Poner VOLUME para permitir configurar la base de datos
# Se define donde se va a ejecutar la base de datos
```

```
VOLUME ["/etc/mysql", "/var/lib/mysql"]

# Se abre el puerto 3306 para permitir la conexión
EXPOSE 3306

# Ejecuta permanentemente mysql
CMD ["/ejecutar.sh"]
```

Como se ve en el contenido del anterior fichero, es necesario modificar el puerto por el que está escuchando la base de datos, es decir, que la máquina que contiene el docker con la base de datos esté escuchando a la IP que contiene el servicio de ownCloud (40.87.121.16), es por eso que en el archivo de configuración de MariaDB (/etc/mysql/my.cnf), se debe modificar la línea `/bind-address`. Por otro, durante la creación de la imagen, es necesario hacer uso de otros dos scripts:

1. `crear_usuario.sh`: crea un usuario en nuestra base de datos.

```
#!/bin/bash

/usr/bin/mysqld_safe > /dev/null 2>&1 &

RET=1
while [[ RET -ne 0 ]]; do
    echo "=> Esperando confirmacion del servicio de MariaDB"
    sleep 5
    mysql -uroot -e "status" > /dev/null 2>&1
    RET=$?
done

PASS=${MARIADB_PASS:-$(pwgen -s 12 1)}
_word=$( [ ${MARIADB_PASS} ] && echo "preset" || echo "random" )
echo "=> Creado usuario"

mysql -uroot -e "CREATE USER 'admin'@'%' IDENTIFIED BY '$PASS'"
mysql -uroot -e "GRANT ALL PRIVILEGES ON *.* TO 'admin'@'%' WITH GRANT OPTION"

echo "=====
mysql -u admin -p $PASS -h <host> -P <port>
=====
mysqladmin -uroot shutdown
```

2. `ejecutar.sh`: ejecuta la instalación de MySQL/MariaDB.

```
#!/bin/bash

VOLUME_HOME="/var/lib/mysql"

if [[ ! -d $VOLUME_HOME/mysql ]]; then
    echo "=> Instalando MariaDB"
    mysql_install_db > /dev/null 2>&1
    echo "=> Instalacion completada"
    /crear_usuario.sh
```

```
else
    echo "=> Usando un volumen existente de MariaDB"
fi

exec mysqld_safe
```

Se deben de crear los tres ficheros dentro de nuestra máquina virtual, en donde a continuación, se debe ejecutar el fichero **Dockerfile**, el cuál instalará la imagen de MariaDB que se acaba de comentar. A continuación, se ponen los pasos necesarios, para el buen funcionamiento de la base de datos. Las siguientes sentencias, deben ser ejecutadas una vez accedido por SSH a la máquina virtual 2.

```
# Instalar docker
sudo apt install docker.io

# Crear imagen para la base de datos
nano Dockerfile
nano ejecutar.sh
nano crear_usuario.sh

# Nos creamos el contenedor
sudo docker build -t gema/mariadb .
sudo docker run -d -p 3306:3306 -e MARIADB_PASS="mypass" gema/mariadb
```

En la variable de entorno **MARIADB_PASS="mypass"**, se debe especificar la contraseña que queremos poner para acceder a la base de datos. Y con esto, ya sólo nos hace falta poner los datos de la base de datos en la página de inicio de ownCloud:

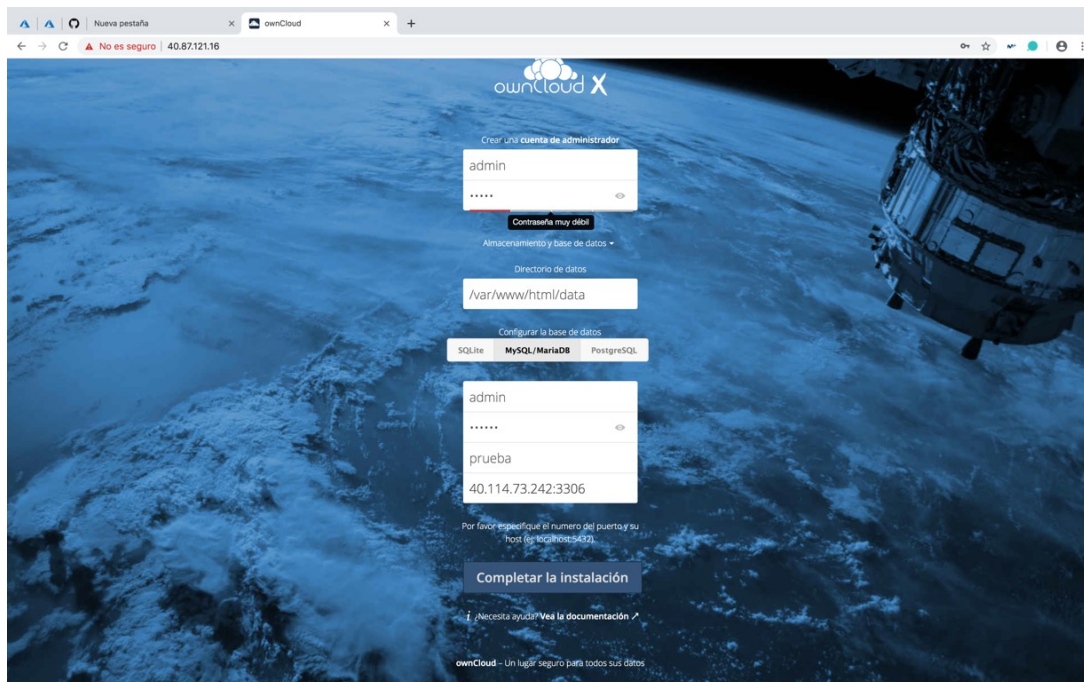


Figura 9: Introduciendo datos de la base de datos

donde:

- *user*: admin
- *pass*: mypass
- *database*: prueba
- *host*: < IP2 >:< puerto > (40.114.73.242:3306)

Al iniciar sesión con el almacenamiento de la base de datos, se nos volverá a pedir el mismo usuario (*admin*) y la contraseña (*admin*).

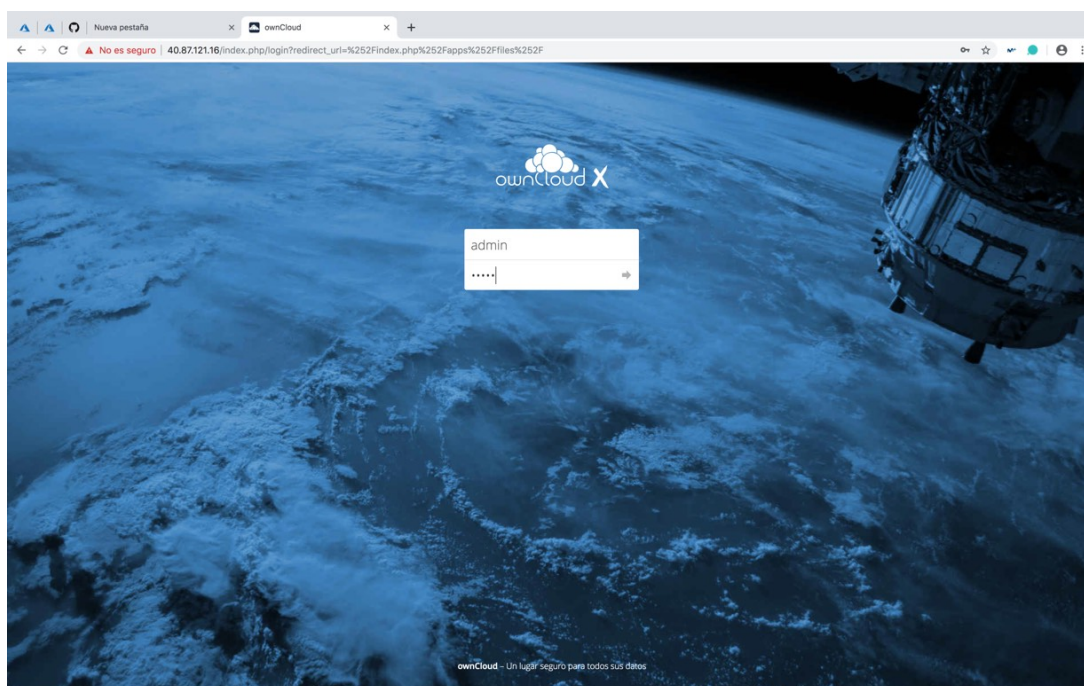


Figura 10: Introduciendo datos de nuevo

3.1.3. Servicio LDAP

Por último, pasamos a la creación de la máquina virtual para **OpenLDAP**, la cual contiene el soporte del sistema de autenticación. Para ello, se ha creado un script `maquina3-ldap.sh` (`sh maquina3-ldap.sh`), el cual nos crea una máquina virtual, con un tamaño especificado y los requerimientos necesarios para poder acceder a dicha máquina virtual.

```
#!/bin/bash

# Creacion de la maquina virtual 3 para LDAP
# Obtenemos la IP
echo " ————— Creacion de la maquina virtual para OpenLDAP ————— "
IP3=$(az vm create ... | jq -r '.publicIpAddress')

# Cambiamos la dimension a la maquina virtual
echo " ————— Cambiando la dimension a la maquina virtual ————— "
```



```
az vm resize --resource-group myResourceGroup-eastus ...

# Abrimos el puerto 389 y 636
echo " ——— Abrimos el puerto 389 y 636 ——— "
az vm open-port --port 389 --resource-group myResourceGroup-eastus ...
az vm open-port --port 636 --resource-group myResourceGroup-eastus ...

# Una vez creada, mostramos su nombre y su direccion IP
echo " ——— Datos de la maquina virtual creada para Owncloud ——— "
echo -name: MV3-ldap
echo -ip: $IP3
```

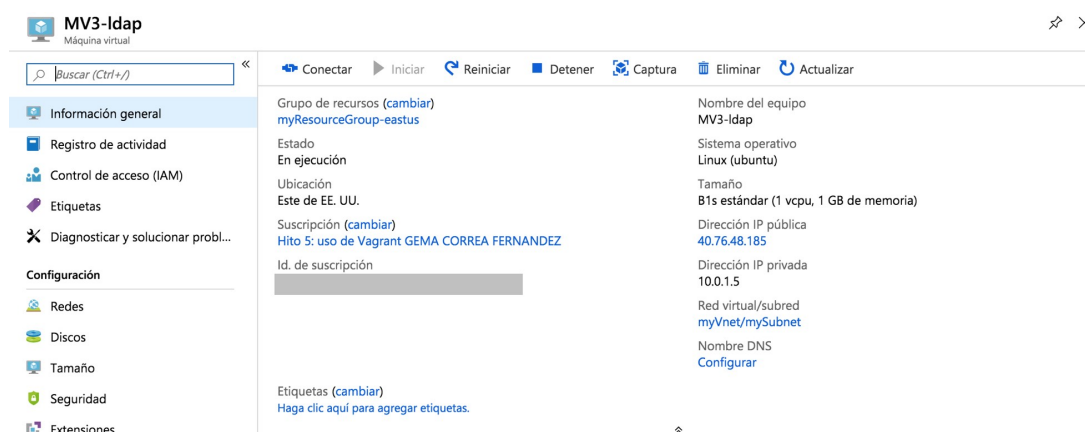


Figura 11: Máquina virtual creada

Podremos encontrar la máquina creada en Azure, con el nombre de MV3-ldap. Escogemos la imagen de *osixia* ya que en prácticas vimos que funcionaba bien, abrimos los puertos 389 y 636 que son los usados tal y como se indica en la documentación de LDAP.

Una vez que hayamos lanzado el script, nos aparecerá en la terminal, una ventana como la siguiente. Así que deberemos escribir en la consola `ssh gemazure@<IP3>` para así poder acceder a ella, se deberá poner la IP que se haya asignado a la máquina, visible desde Azure o desde el terminal. Una vez creada la máquina virtual y con acceso a ella, instalaremos el contenedor correspondiente a *osixia*.

```
# Conectarnos a la maquina virtual
ssh gemazure@$IP3

# Instalar docker y ldap
sudo apt install docker.io
sudo apt-get update
sudo apt install ldap-utils

# Nos descargamos openldap
sudo docker pull osixia/openldap
sudo docker run -d -p 389:389 --name ldap -t osixia/openldap
```

```

----- Datos de la máquina virtual creada para Owncloud -----
-name: MV3-ldap
-ip: 40.76.48.185
MacBook-Pro-de-Gema:Práctica 1 gema$ ssh gemazure@40.76.48.185
The authenticity of host '40.76.48.185 (40.76.48.185)' can't be established.
ECDSA key fingerprint is SHA256:GGzq3LzGfUgP9UKUVj/jfjCRnt39MwhnnX54E2UCVio.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '40.76.48.185' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.18.0-1014-azure x86_64)

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

System information as of Sat May  4 12:31:43 UTC 2019

System load:  0.06          Processes:            112
Usage of /:   4.1% of 28.90GB Users logged in:       0
Memory usage: 25%          IP address for eth0: 10.0.1.5
Swap usage:   0%

2 packages can be updated.
2 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

gemazure@MV3-ldap:~$

```

Figura 12: Accediendo a la máquina virtual por SSH

Cuando ya disponemos de nuestro contenedor, pasamos a la creación de usuarios:

```

# Crear fichero para el usuario y el grupo de usuario
nano gema.ldif

# Aadir usuario a LDAP
ldapadd -H ldap://40.76.48.185 -x -D "cn=admin,dc=example,dc=org" -w
admin -c -f gema.ldif

# Cambiar el password
ldappasswd -s admin -W -D "cn=admin,dc=example,dc=org" -x "cn=gema,ou=
users,dc=example,dc=org"

gemazure@MV3-ldap:~$ ldapadd -H ldap://40.76.48.185 -x -D "cn=admin,dc=example,dc=org" -w admin -c -f gema.ldif
adding new entry "ou=users,dc=example,dc=org"

adding new entry "ou=groups,dc=example,dc=org"

adding new entry "cn=practica1,ou=groups,dc=example,dc=org"

adding new entry "cn=gema,ou=users,dc=example,dc=org"

modifying entry "cn=practica1,ou=groups,dc=example,dc=org"

```

Figura 13: Añadiendo usuario y grupo de usuarios

A continuación, mostramos el contenido de la creación de un usuario llamado *gema*, perteneciente al grupo de usuarios *practica1*, con la contraseña *admin*:

```
dn: ou=users , dc=example ,dc=org
ou: users
description: All people in organisation
objectclass: organizationalunit

dn: ou=groups , dc=example ,dc=org
ou: groups
description: All Groups in organisation
objectclass: organizationalunit

dn: cn=practical ,ou=groups ,dc=example ,dc=org
objectClass: groupOfUniqueNames
cn: practical
uniqueMember: cn=practical ,ou=groups ,dc=example ,dc=org
description: practical

dn: cn=gema,ou=users ,dc=example ,dc=org
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: gema
uid: gema
uidNumber: 16859
gidNumber: 100
homeDirectory: /home/gema
loginShell: /bin/bash
gecos: gema
userPassword: {crypt}x
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0

dn: cn=practical ,ou=groups ,dc=example ,dc=org
changetype: modify
add: uniqueMember
uniqueMember: cn=gema ,ou=users ,dc=example ,dc=org
```

Con esto ya tenemos el servidor LDAP funcionando y con un usuario creado.

3.2. Configuración de LDAP en ownCloud

Teniendo los tres servicios en correcto funcionamiento, vamos a realizar la configuración de nuestra aplicación:

1. Accedemos a ownCloud (<http://40.87.121.16/>), y ponemos los datos que hemos comentado previamente.
2. Dentro de ownCloud, nos dirigimos a Market y buscamos LDAP Integration, para su posterior instalación, dicha aplicación delega la gestión de usuarios a un servidor LDAP.
3. Una vez instalada, vamos a Ajustes>Autenticación de usuario y allí introducimos los datos del servidor LDAP tal y como aparece en la siguiente imagen (hay

que tener el docker de LDAP funcionando previamente).

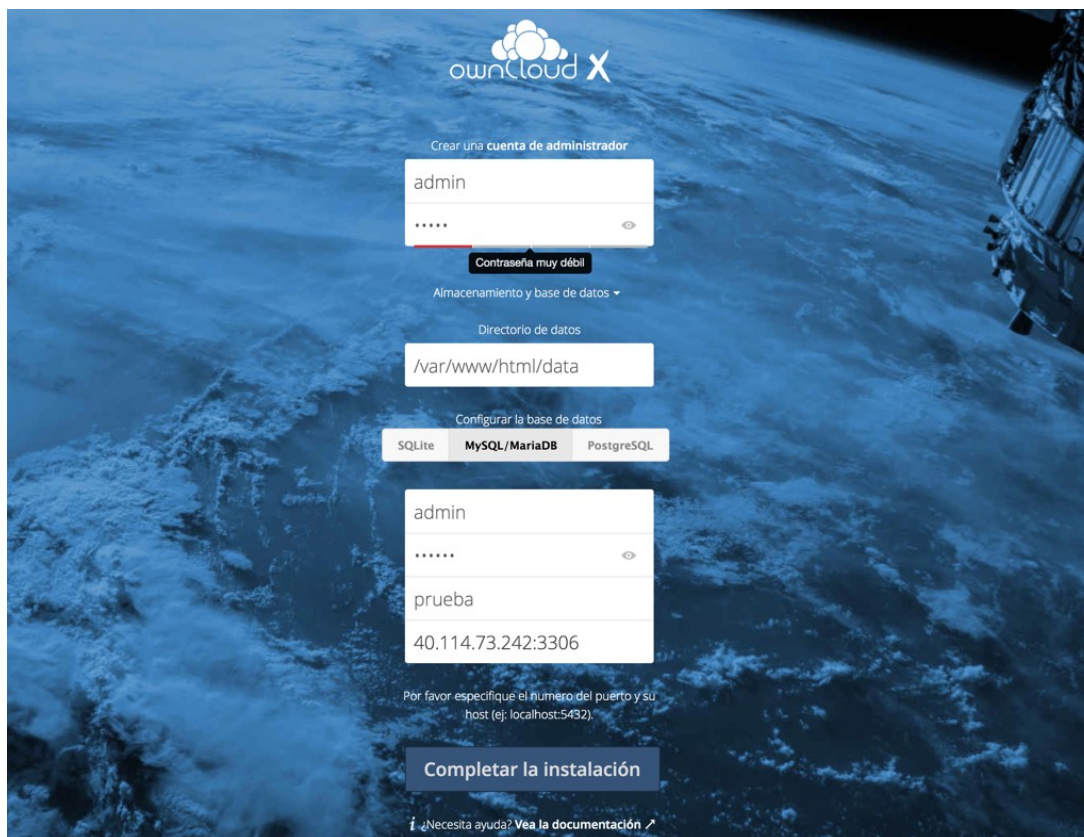
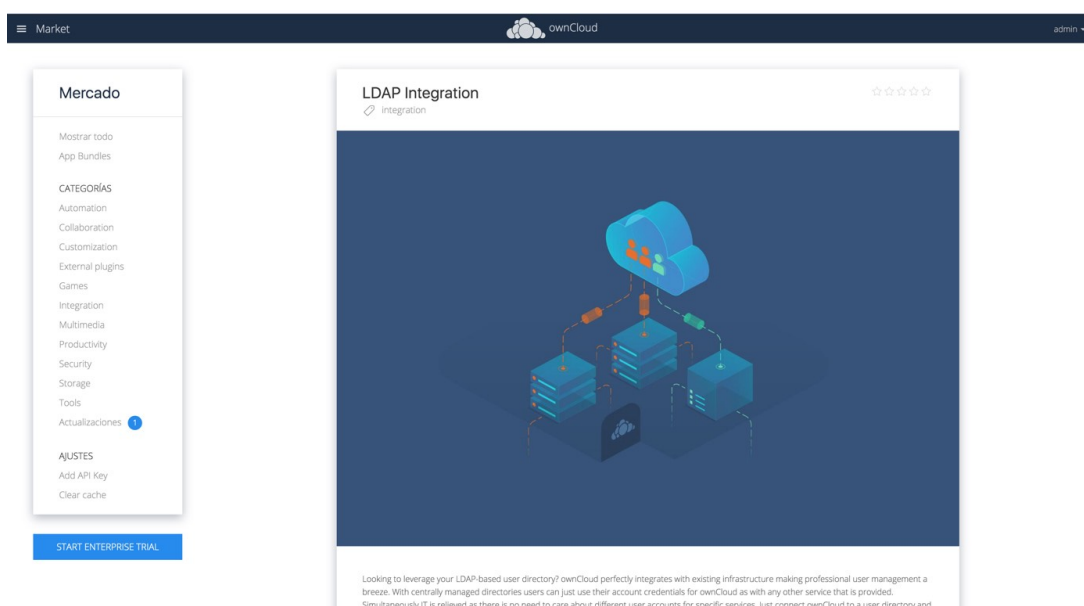
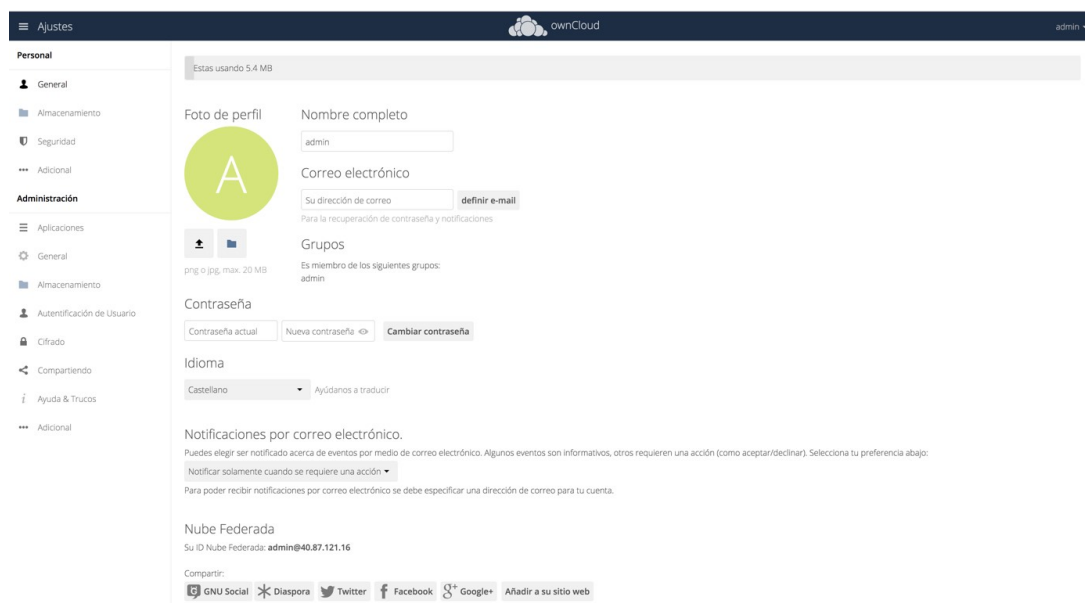


Figura 14: Acceder a ownCloud



Looking to leverage your LDAP-based user directory? ownCloud perfectly integrates with existing infrastructure making professional user management a breeze. With centrally managed directories users can just use their account credentials for ownCloud as with any other service that is provided. Simultaneously IT is relieved as there is no need to care about different user accounts for specific services. Just connect ownCloud to a user directory and

Figura 15: Instalar LDAP Integration



ownCloud

admin

Ajustes

Personal

Estas usando 5.4 MB

Foto de perfil

Nombre completo

admin

Correo electrónico

Su dirección de correo definir e-mail

Para la recuperación de contraseña y notificaciones:

Grupos

Es miembro de los siguientes grupos:

admin

Contraseña

Contraseña actual Nueva contraseña Cambiar contraseña

Idioma

Castellano Ayúdanos a traducir

Notificaciones por correo electrónico.

Puedes elegir ser notificado acerca de eventos por medio de correo electrónico. Algunos eventos son informativos, otros requieren una acción (como aceptar/declinar). Selecciona tu preferencia abajo:

Notificar solamente cuando se requiere una acción

Para poder recibir notificaciones por correo electrónico se debe especificar una dirección de correo para tu cuenta.

Nube Federada

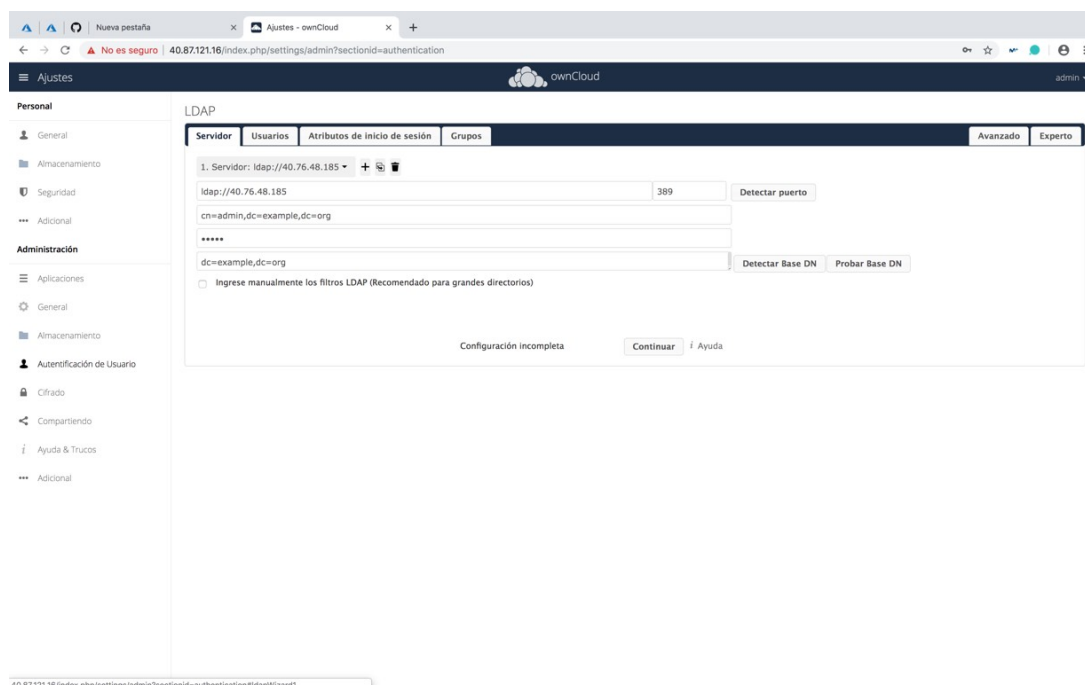
Su ID Nube Federada: admin@40.87.121.16

Compartir:

GNU Social Diaspora Twitter Facebook Google+ Añadir a su sitio web

Figura 16: Ajustes > Autenticación de usuario

4. Introducimos la IP del docker de LDAP y el puerto usado (ldap://40.76.48.185), si se pulsa detectar puerto debería aparecer el puerto 389. Introducimos el DIT `cn=admin,dc=example,dc=org`. La contraseña por defecto es `admin` y la base DN es `dc=example,dc=org`. Si todo funciona correctamente, abajo debería aparecer un círculo verde e indicar que la configuración es correcta.



ownCloud

admin

Ajustes

Personal

LDAP

Servidor Usuarios Atributos de inicio de sesión Grupos Avanzado Experto

1. Servidor: ldap://40.76.48.185 + -

ldap://40.76.48.185 389 Detectar puerto

cn=admin,dc=example,dc=org

dc=example,dc=org Detectar Base DN Probar Base DN

☐ Ingrese manualmente los filtros LDAP (Recomendado para grandes directorios)

Configuración incompleta Continuar Ayuda

Figura 17: Configuración paso I

5. Seguimos avanzando por las pestañas, y seleccionamos todas las posibles opciones, en donde seleccionamos el grupo de usuarios que hemos creado (*practica1*):

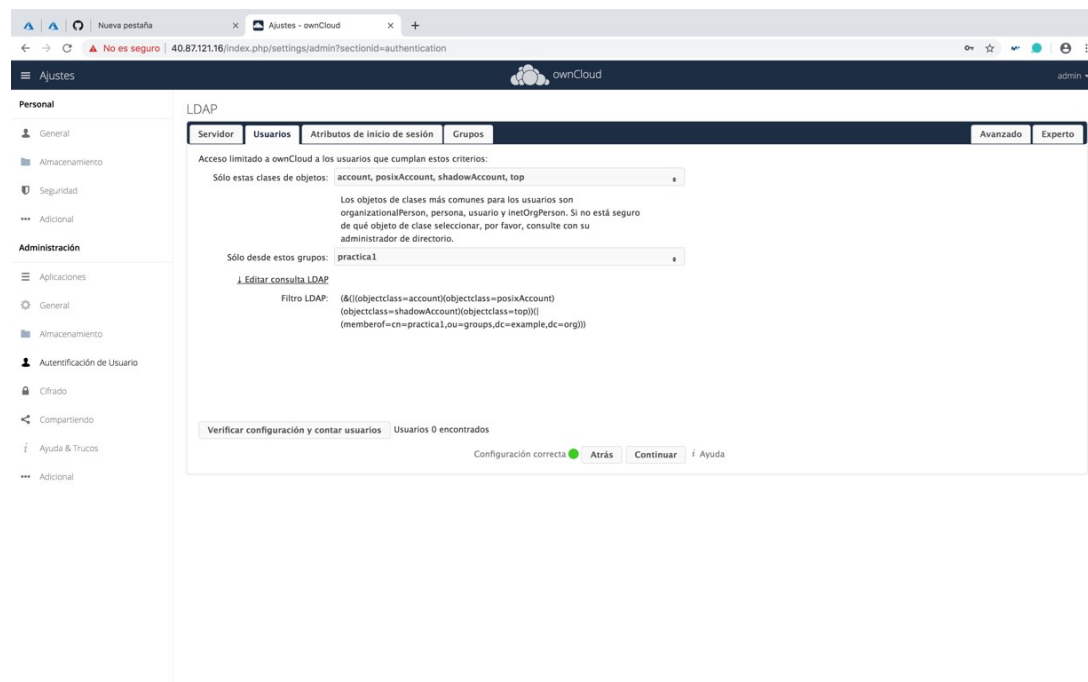


Figura 18: Configuración paso II

6. Ponemos que por defecto, para iniciar sesión coja los campos *cn* y *userPassword*. En nuestro caso, hemos creado un usuario *gema* y su contraseña es *admin*.

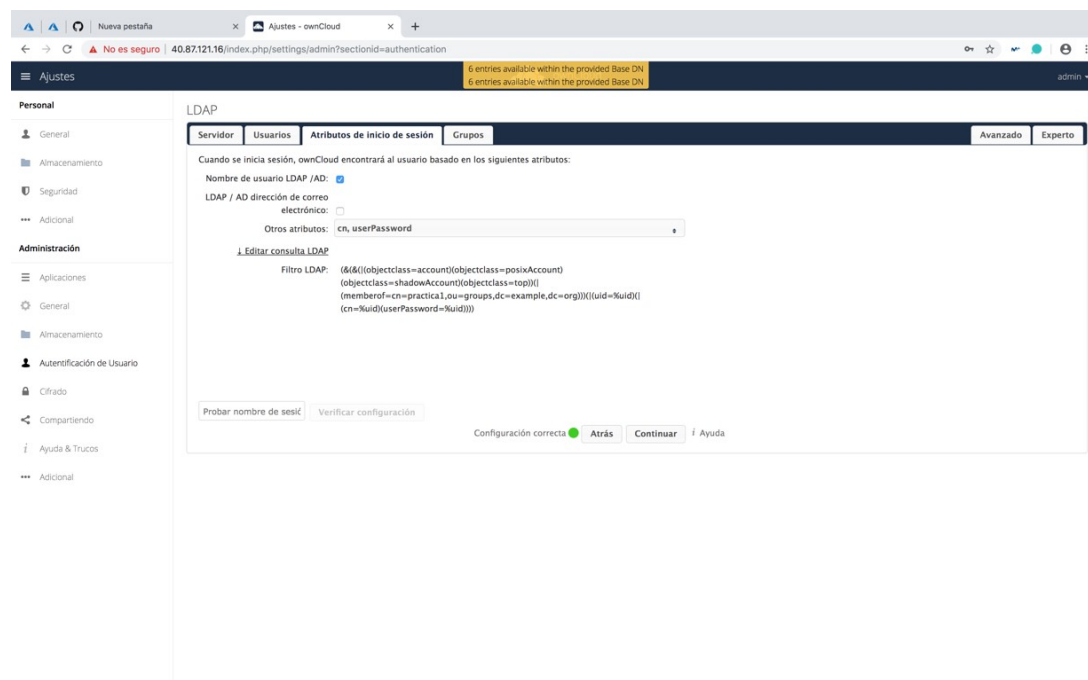


Figura 19: Configuración paso III

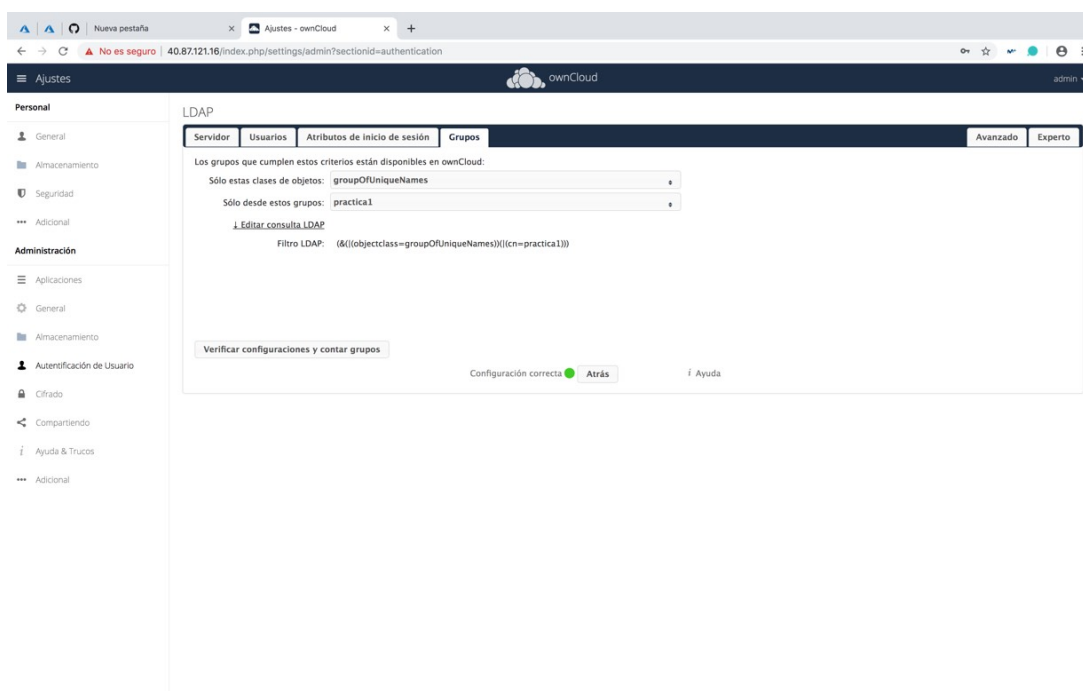


Figura 20: Configuración paso IV

Con esto terminamos la configuración de LDAP. Si salimos e iniciamos sesión con el usuario creado con el archivo `gema.ldif` y con la contraseña que establecimos con el comando `ldappasswd`, ahora debería aparecer en la pestaña de usuarios el usuario administrador y el usuario creado en LDAP.

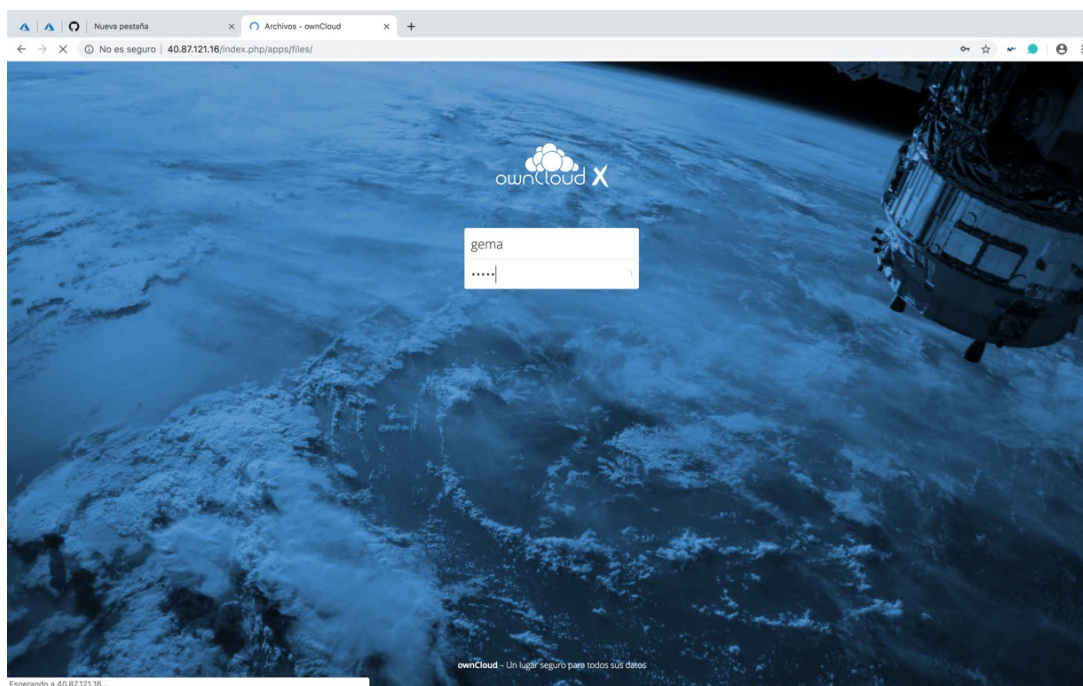


Figura 21: Accediendo con el nuevo usuario

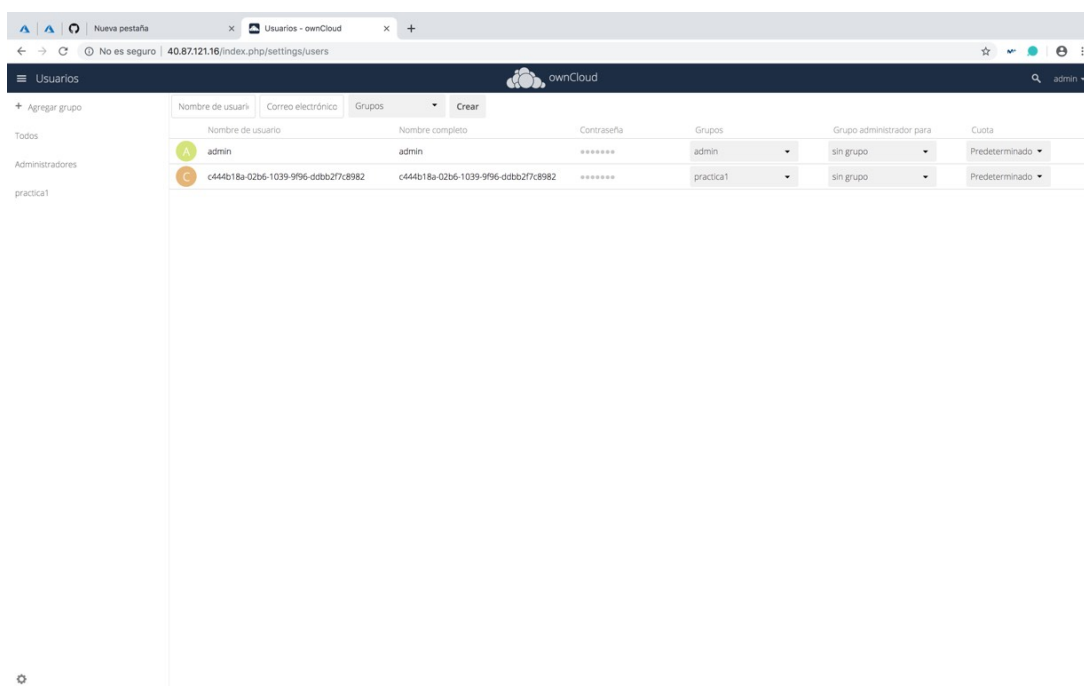


Figura 22: Aparece el usuario creado en la sesión del admin

3.3. Comprobar conexión ownCloud y MariaDB

Puesto que se acaba de comprobar la conexión de LDAP y ownCloud, vamos a ver la conexión que existe entre ownCloud y MariaDB. Para ello, vamos a intentar almacenar un archivo cualquiera en nuestros documentos para comprobar el funcionamiento de ownCloud. Para ello subimos el archivo a la interfaz de ownCloud.

```
mysql> SELECT User FROM mysql.user;
+-----+
| User |
+-----+
| admin |
| oc_admin |
| root |
| root |
| root |
| root |
| oc_admin |
| root |
+-----+
7 rows in set (0,12 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| prueba |
+-----+
4 rows in set (0,12 sec)

mysql> show tables;
+-----+
| Tables_in_prueba |
+-----+
| oc_account_terms |
| oc_accounts |
| oc_addressbookchanges |
| oc_addressbooks |
| oc_appconfig |
| oc_auth token |
+-----+
```

Figura 23: Comprobando acceso a ownCloud

```
# Para comprobar conexión
mysql -u admin -p prueba -h 40.114.73.242 -P 3306
# y meto la contraseña "mypass"
SELECT User FROM mysql.user;
show databases;
show tables;
```

4. Manual de usuario

En este apartado se indican los pasos necesarios para reproducir la práctica. Se debe tener en cuenta, que la configuración de LDAP en ownCloud es la misma que la comentada en el despliegue. Primero hay que clonar el repositorio de la práctica (https://github.com/Gecofer/MII_CC2_1819).

1. Instalar si no se tiene el CLI de Azure y hacer `az login`
2. Ejecutar el script `maquina1-owncloud.sh`
3. Ejecutar el script `maquina2-database.sh`
4. Ejecutar el script `maquina3-ldap.sh`
5. Acceder por SSH a las tres máquinas creadas, en pestañas de consola distintas:

```
# Maquina 1 - owncloud
ssh gemazure@40.87.121.16

# Maquina 2 - mariadb
ssh gemazure@40.114.73.242

# Maquina 3 - ldap
ssh gemazure@40.76.48.185
```

6. En el terminal de la máquina 1:

```
# Instalar docker
sudo apt install docker.io

# Descargar owncloud y ponerlo en funcionamiento
sudo docker pull owncloud:latest
sudo docker run -d -p 80:80 owncloud:latest
```

7. En el terminal de la máquina 2:

```
# Instalar docker
sudo apt install docker.io

# Crear imagen para la base de datos
```

```
nano Dockerfile
nano ejecutar.sh
nano crear_usuario.sh

# Nos creamos el contenedor
sudo docker build -t gema/mariadb .
sudo docker run -d -p 3306:3306 -e MARIADB_PASS="mypass" gema/
mariadb
```

8. En el terminal de la máquina 3:

```
# Instalar docker y ldap
sudo apt install docker.io
sudo apt-get update
sudo apt install ldap-utils

# Nos descargamos openldap
sudo docker pull osixia/openldap
sudo docker run -d -p 389:389 --name ldap -t osixia/openldap
```

9. Entramos al navegador y ponemos la IP de la máquina que contiene el servicio de ownCloud.
10. Añadimos los datos del usuarios y de la base de datos.
11. Configuramos LDAP en ownCloud de la misma manera que hemos explicado en el despliegue.
12. Si queremos añadir un nuevo usuario que se llame *antonio* y pertenezca al grupo *practical1*, nos creamos un archivo **antonio.ldif** que contenga:

```
dn: cn=antonio,ou=users,dc=example,dc=org
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: antonio
uid: antonio
uidNumber: 16859
gidNumber: 100
homeDirectory: /home/antonio
loginShell: /bin/bash
gecos: antonio
userPassword: {crypt}x
shadowLastChange: 0
shadowMax: 0
shadowWarning: 0

dn: cn=practical1,ou=groups,dc=example,dc=org
changetype: modify
add: uniqueMember
uniqueMember: cn=antonio,ou=users,dc=example,dc=org
```

```
# Crear fichero
antonio.ldif

# Insertar usuario
ldapadd -H ldap://40.76.48.185 -x -D "cn=admin,dc=example,dc=org" -w
admin -c -f antonio.ldif

# Cambiar password
ldappasswd -s admin -W -D "cn=admin,dc=example,dc=org" -x "cn=
antonio,ou=users,dc=example,dc=org"
# Ahora es antonio-admin
```

5. Mejoras

Se ha introducido como mejora, la automatización de todos los comandos necesarios para la creación de las máquinas virtuales en Azure, mediante el uso de scripts de shell. Además, toda la información necesaria para ejecutar la aplicación se encuentra en el repositorio https://github.com/Gecofer/MII_CC2_1819.

Por otro lado, cabe destacar que en vez de usar scripts de shell, se podría haber hecho uso de la integración de Vagrant con Azure junto con el aprovisionamiento de Ansible para la orquestación.

Referencias

- [1] *Virtualización basada en contenedores (virtualización a nivel de sistema operativo)*, SearchDataCenter Tchtarget, dirección: <https://searchdatacenter.techtarget.com/es/definicion/virtualizacion-basada-en-contenedores-virtualizacion-a-nivel-de-sistema-operativo> (visitado 5 de mayo de 2019).
- [2] *ownCloud vs. Nextcloud: almacenamiento en la nube comparado, 11 IONOS*, dirección: <https://www.ionos.es/digitalguide/servidores/herramientas/owncloud-vs-nextcloud-ventajas-e-inconvenientes/> (visitado 5 de mayo de 2019).
- [3] *The Difference Between Active Directory and LDAP*, Varonis, dirección: <https://www.varonis.com/blog/the-difference-between-active-directory-and-ldap/> (visitado 5 de mayo de 2019).
- [4] *Escalado automático de máquinas virtuales en Azure*, Microsoft Azure, dirección: <https://docs.microsoft.com/es-es/azure/virtual-machines/windows/autoscale> (visitado 5 de mayo de 2019).
- [5] *MariaDB version*, Github, dirección: <https://github.com/katafractari/mariadb-owncloud-docker> (visitado 5 de mayo de 2019).
- [6] *Escalabilidad de Maquinas Virtuales en Microsoft Azure*, Blog JGAITProConocimiento para TI, dirección: <https://blog.jgaitpro.com/escalabilidad-de-maquinas-virtuales-en-microsoft-azure/> (visitado 5 de mayo de 2019).
- [7] *What are the differences between LDAP and Active Directory?*, Stackoverflow, dirección: <https://stackoverflow.com/questions/663402/what-are-the-differences-between-ldap-and-active-directory> (visitado 5 de mayo de 2019).