

# Docker compose

## Docker

Fuente: <https://docs.docker.com/engine/install/ubuntu/>

Quitamos dependencias antiguas

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do sudo apt-get remove $pkg; done
```

Actualizamos repositorios

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

Instalamos docker

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Usuario sin sudo

```
sudo vi /etc/group
docker:x:988:soffid01
```

```
sudo systemctl restart docker
```

```
sudo usermod -aG docker soffid01
```

```
exit > login
```

```
docker ps
```

# Docker compose

Documentación Soffid: <https://bookstack.soffid.com/books/installation/page/installing-soffid>

Documentación Docker compose: <https://docs.docker.com/reference/cli/docker/compose/>

```
sudo vi docker-compose.yaml
```

```
services:
  mariadb:
    image: mariadb:11.1.2
    environment:
      MYSQL_ROOT_PASSWORD: dkF45.r4f
      MYSQL_DATABASE: soffid
      MYSQL_USER: soffid
      MYSQL_PASSWORD: 98nds.D3
    ports:
      - 3306:3306
    healthcheck:
      test: "/usr/bin/mariadb --user=root --password=dkF45.r4f --execute \"SHOW DATABASES;\""
      interval: 2s
      timeout: 20s
      retries: 10
    command:
      - --max_allowed_packet=128M
      - --innodb_log_file_size=256M
      - --character-set-server=utf8mb4
      - --collation-server=utf8mb4_general_ci
      - --server-id=1
```

- --log-bin
- --binlog-format=row
- --expire-logs-days=15
- --max-binlog-size=1000M
- --replicate-ignore-table=soffid.SC\_SEQUENCE
- --slave-skip-errors=1032,1053,1062

networks:

- network

volumes:

- mariadb\_data:/var/lib/mysql

console:

image: soffid/iam-console:3.6.17

environment:

DB\_URL: jdbc:mariadb://mariadb/soffid

DB\_USER: soffid

DB\_PASSWORD: 98nds.D3

ports:

- 8080:8080

networks:

- network

healthcheck:

test: bash -c "(echo 'GET /soffid/anonymous/logo.svg HTTP/1.1' >&0; echo >&0; cat >&2;) <>

/dev/tcp/localhost/8080"

interval: 10s

timeout: 20s

retries: 10

start\_period: 40s

volumes:

- console\_trust:/opt/soffid/iam-console-3/trustedcerts
- console\_conf:/opt/soffid/iam-console-3/conf
- console\_logs:/opt/soffid/iam-console-3/logs
- console\_index:/opt/soffid/iam-console-3/index

depends\_on:

mariadb:

condition: service\_healthy

sync-server:

image: soffid/iam-sync:3.6.14

hostname: soffid01

```
environment:
  SOFFID_PORT: 1760
  SOFFID_HOSTNAME: soffid01
  SOFFID_MAIN: yes
  DB_URL: jdbc:mysql://mariadb/soffid
  DB_USER: soffid
  DB_PASSWORD: 98nds.D3
ports:
  - 1760:1760
networks:
  - network
volumes:
  - sync_conf:/opt/soffid/iam-sync/conf
depends_on:
  mariadb:
    condition: service_healthy
  console:
    condition: service_healthy
```

```
networks:
  network:
    name: netcompose
    driver: bridge
```

```
volumes:
  mariadb_data:
    name: compose_mariadbdata
  console_trust:
    name: compose_console_trustedcerts
  console_conf:
    name: compose_console_conf
  console_logs:
    name: compose_console_logs
  console_index:
    name: compose_console_index
  sync_conf:
    name: compose_sync_conf
```

En la máquina 2 hay un campo diferentes (ver sección réplica base de datos).

```
- --server-id=2
```

...

sync-server:

image: soffid/iam-sync:3.6.14

hostname: soffid02

environment:

SOFFID\_PORT: 1760

SOFFID\_HOSTNAME: soffid02

Iniciamos los contenedores.

```
docker compose up -d
```

Comandos útiles.

```
docker compose ps
```

```
docker compose logs -f console
```

```
docker compose logs -f sync-server
```

```
docker compose exec -it console bash
```

```
docker compose exec -it sync-server bash
```

# Notas sobre el yaml

Aquí incluyo una explicación de los puntos importantes del yaml

```
// Se ha añadido para que levante los servicios si hay un reinicio del
```

```
// servicio docker o de la máquina.
```

```
// Añadido en cada uno de los servicios.
```

```
restart: always
```

```
// Se añadido para las réplicas de la base de datos.
```

```
// Cada servidor tiene un server-id diferente (el 1 y el 2)
```

```
- --server-id=1
```

```
- --log-bin
```

```
- --binlog-format=row
- --expire-logs-days=15
- --max-binlog-size=1000M
- --replicate-ignore-table=soffid.SC_SEQUENCE
- --slave-skip-errors=1032,1053,1062
```

```
// Las bases de datos, aunque sean replicas, la primera actua de principal
// y solo se utiliza la segunda en caso de caída de la primera.
// Se utilizad una configuración en modo secuencial
```

```
DB_URL: jdbc:mariadb:sequential://172.16.9.20,172.16.9.35/soffid
```

```
// El hostname del syncserver tiene que ser el nombre de la máquina
```

```
hostname: soffid01
```

```
SOFFID_HOSTNAME: soffid01
```

```
// Para la configuración del ldap, el usuario requiere incluir el "cn="
```

```
USER: cn=admin
```

## Habilitar port forwarding

Primero hay que confirmar o actualizar la configuración del port forwarding del servidor.

```
sudo vi /etc/ssh/sshd_config

AllowTcpForwarding yes

sudo systemctl restart ssh
```

Abrimos el puerto por port forwarding mediante una conexión ssh.

```
ssh -L 8080:localhost:8080 soffid01@172.16.9.20
```

Ahora accedemos a través del navegador.

http://localhost:8080

Usuarios admin y svives creados y con password en el vault.

# Replica de BBDD

Documentación Soffid: <https://bookstack.soffid.com/books/installation/page/creating-a-multimaster-mariadb-replica-2b4>

Documentación EMASA: <https://bookstack.soffid.com/books/emas/page/sincronizar-bases-de-datos>

Paso 1: actualizar la configuración del yaml para incluir parámetros de configuración de Mariadb

Añadir estos parámetros en el servidor 1.

```
- --server-id=1
- --log-bin
- --binlog-format=row
- --expire-logs-days=15
- --max-binlog-size=1000M
- --replicate-ignore-table=soffid.SC_SEQUENCE
- --slave-skip-errors=1032,1053,1062
```

En el servidor 2.

```
- --server-id=2
- --log-bin
- --binlog-format=row
- --expire-logs-days=15
- --max-binlog-size=1000M
- --replicate-ignore-table=soffid.SC_SEQUENCE
- --slave-skip-errors=1032,1053,1062
```

Paso 2: tenemos que clonar la base de datos 1 en la 2

Hacemos un backup de la base de datos del servidor 1.

```
docker compose exec -it mariadb bash -c 'mariadb-dump -u root --password="dkF45.r4f" soffid' > mariadb-backup.sql
```

Copiamos el fichero al servidor 2.

```
scp mariadb-backup.sql soffid02@172.16.9.35:/home/soffid02/
```

Cargamos el backup en el servidor 2.

```
docker compose cp mariadb-backup.sql mariadb:/tmp
docker compose exec -it mariadb bash
mariadb -u soffid -p soffid < /tmp/mariadb-backup.sql
```

### Paso 3: creamos los usuarios de base de datos que ejecutarán las réplicas

Creamos en el servidor 1 un usuario para replicar los datos que usará el servidor 2.

NOTA: es importante que las contraseñas sean las mismas porque las replicas sobre-escribirán el usuario.

```
docker compose exec -it mariadb bash
mariadb -u root --password="$MYSQL_ROOT_PASSWORD"
create user replication_user@172.16.9.35 identified by 'SDfh.343';
grant replication slave on *.* to replication_user@172.16.9.35;
set password for replication_user@172.16.9.35 = password('SDfh.343');
```

Y en el servidor 1.

```
docker compose exec -it mariadb bash
mariadb -u root --password="$MYSQL_ROOT_PASSWORD"
create user replication_user@172.16.9.20 identified by 'SDfh.343';
grant replication slave on *.* to replication_user@172.16.9.20;
set password for replication_user@172.16.9.20 = password('SDfh.343');
```

### Paso 4: creamos de los procesos slave

Consultamos la base de datos 1.

```
MariaDB [soffid]> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 | 1335314 |              |                  |
+-----+-----+-----+-----+
```

Configuramos la replica en la base de datos 2.



```
CHANGE MASTER TO
MASTER_HOST='172.16.9.20',
MASTER_USER='replication_user',
MASTER_PASSWORD='Adfv45.d',
MASTER_PORT=3306,
MASTER_LOG_FILE='mysqld-bin.000001',
MASTER_LOG_POS=1335314,
MASTER_CONNECT_RETRY=10;
```

Consultamos si está activo.

```
MariaDB [(none)]> SHOW SLAVE STATUS \G
***** 1. row *****

Slave_IO_State: Waiting for master to send event
Master_Host: 172.16.9.35
Master_User: replication_user
Master_Port: 3306
Connect_Retry: 10
Master_Log_File: mysqld-bin.000002
Read_Master_Log_Pos: 1408
Relay_Log_File: mysqld-relay-bin.000002
Relay_Log_Pos: 1485
Relay_Master_Log_File: mysqld-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Rewrite_DB:
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table: soffid.SC_SEQUENCE
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 1408
Relay_Log_Space: 1795
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
```

```
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 0
Last_IO_Error:
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 2
Master_SSL_Crl:
Master_SSL_Crlpath:
Using_Gtid: No
Gtid_IO_Pos:
Replicate_Do_Domain_Ids:
Replicate_Ignore_Domain_Ids:
Parallel_Mode: optimistic
SQL_Delay: 0
SQL_Remaining_Delay: NULL
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
Slave_DDL_Groups: 3
Slave_Non_Transactional_Groups: 0
Slave_Transactional_Groups: 2
1 row in set (0.001 sec)
```

Si no funciona, podemos volver a arrancarlo (o pararlo).

```
start slave;
stop slave;
```

Cuando funcione aparecerá esto.

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
```

Ahora lo hacemos en la máquina 1.

Consultamos el estado de la maquina 2.

```
MariaDB [(none)]> show master status;
```

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysqld-bin.000002 | 343 |           |           |
+-----+-----+-----+-----+
```

Iniciamos replica en el servidor 1.

```
CHANGE MASTER TO
MASTER_HOST='172.16.9.35',
MASTER_USER='replication_user',
MASTER_PASSWORD='SDfh.343',
MASTER_PORT=3306,
MASTER_LOG_FILE='mysqld-bin.000002',
MASTER_LOG_POS=343,
MASTER_CONNECT_RETRY=10;
```

Consultamos si está activo.

```
MariaDB [(none)]> SHOW SLAVE STATUS \G
```

```
***** 1. row *****
```

```
Slave_IO_State: Waiting for master to send event
Master_Host: 172.16.9.20
Master_User: replication_user
Master_Port: 3306
Connect_Retry: 10
Master_Log_File: mysqld-bin.000002
Read_Master_Log_Pos: 1181913
Relay_Log_File: mysqld-relay-bin.000004
Relay_Log_Pos: 682878
Relay_Master_Log_File: mysqld-bin.000002
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Replicate_Rewrite_DB:
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
```

Replicate\_Ignore\_Table: soffid.SC\_SEQUENCE

Replicate\_Wild\_Do\_Table:

Replicate\_Wild\_Ignore\_Table:

Last\_Errno: 0

Last\_Error:

Skip\_Counter: 0

Exec\_Master\_Log\_Pos: 1181913

Relay\_Log\_Space: 6085768

Until\_Condition: None

Until\_Log\_File:

Until\_Log\_Pos: 0

Master\_SSL\_Allowed: No

Master\_SSL\_CA\_File:

Master\_SSL\_CA\_Path:

Master\_SSL\_Cert:

Master\_SSL\_Cipher:

Master\_SSL\_Key:

Seconds\_Behind\_Master: 0

Master\_SSL\_Verify\_Server\_Cert: No

Last\_IO\_Errno: 0

Last\_IO\_Error:

Last\_SQL\_Errno: 0

Last\_SQL\_Error:

Replicate\_Ignore\_Server\_Ids:

Master\_Server\_Id: 1

Master\_SSL\_Crl:

Master\_SSL\_Crlpath:

Using\_Gtid: No

Gtid\_IO\_Pos:

Replicate\_Do\_Domain\_Ids:

Replicate\_Ignore\_Domain\_Ids:

Parallel\_Mode: optimistic

SQL\_Delay: 0

SQL\_Remaining\_Delay: NULL

Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates

Slave\_DDL\_Groups: 0

Slave\_Non\_Transactional\_Groups: 0

Slave\_Transactional\_Groups: 15288

1 row in set (0.001 sec)

### Paso 5: actualizamos la tabla de secuencias

Si no hemos iniciado la Consola hay que crear la tabla SC\_SECUENCIAS, pero en este caso ya existía por haber arrancado la Consola, en este caso hay tabla con un registro superior al anterior (incluyendo la suma de la caché), uno impar, el otro par, y ambos que aumenten de dos en dos.

```
DELETE FROM SC_SEQUENCE WHERE SEQ_NEXT=31801;
```

```
DELETE FROM SC_SEQUENCE WHERE SEQ_NEXT=16201;
```

```
INSERT INTO SC_SEQUENCE VALUES (32000, 100, 2);
```

```
INSERT INTO SC_SEQUENCE VALUES (32001, 100, 2);
```

(mejor hacerlo con un UPDATE con un única sql)

### Paso 6: acceso a las bases de datos de forma secuencial

Hay que cambiar la cadena de conexión de las Consolas y Syncservers para apuntar primero a la base de datos 1 y en caso de fallo que vaya a la base de datos 2.

```
jdbc:mariadb:sequential://172.16.9.20,172.16.9.35/soffid
```

NOTA: hemos movido la base de datos a /soffid porque no había espacio

```
# No se paraban
```

```
docker compose ps
```

```
docker compose stop soffid-ldap
```

```
docker compose stop sync-server
```

```
docker compose stop console
```

```
docker compose stop mariadb
```

```
# No se paraban
```

```
docker ps
```

```
docker stop soffid01-soffid-ldap-1
```

```
docker stop soffid01-sync-server-1
```

```
docker stop soffid01-console-1
```

```
docker stop soffid01-mariadb-1
```

```
docker ps
```

```
docker rm -f soffid01-soffid-ldap-1
```

```
docker rm -f soffid01-sync-server-1
```

```
docker rm -f soffid01-console-1
```

```
docker rm -f soffid01-mariadb-1
```

```
ls /var/lib/mysql
```

```
ls -ldrw-x--x--- 12 root root 4096 Dec 2 16:39 docker/
```

```
sudo systemctl status docker
```

```
sudo systemctl stop docker
```

```
sudo systemctl stop docker.socket
```

```
sudo mv /var/lib/docker /soffid/docker
```

```
sudo ln -s /soffid/docker /var/lib/docker
```

```
df -h
```

```
sudo systemctl start docker
```

```
sudo systemctl status docker
```

```
sudo systemctl start docker.socket
```

# Certificados

Los certificados están en la máquina 1 en la carpeta "/ssl-2024/".

Pruebas para validar que podemos abrir el certificado.

```
docker compose cp /ssl-2024/unal.pfx console:/opt/soffid/iam-console-3/trustedcerts/
```

```
docker compose exec -it console /bin/bash
```

```
cd /opt/soffid/iam-console-3/trustedcerts/
```

```
keytool -v -list -keystore unal.pfx
```

# Balanceador

Pendientes de saber si hay un balanceador en la infraestructura.

## Soffid LDAP

Documentación: <https://bookstack.soffid.com/books/soffid-ldap/page/how-to-install-soffid-ldap>

Creado primero como un fichero independiente, luego se mergeará con el anterior (docker-compose.yaml.backup).

```
services:
  soffid-ldap:
    image: soffid/soffidldap:15
    environment:
      SOFFID_SERVER: https://sync-server.netcompose:1760
      SOFFID_AGENT: ldapSoffid01
      USER: cn=admin
      PASSWORD: RTZlv6EkNACdN7xsd4jVRt3D
      DN: o=unal.edu.co
    ports:
      - 389:389
      - 636:636
    networks:
      - network
    volumes:
      - ldapconf:/etc/ldap/slapd.d
      - ldapdata:/var/lib/ldap

networks:
  network:
    name: netcompose
    driver: bridge
```

volumes:

ldapconf:

name: compose\_ldapconf

ldapdata:

name: compose\_ldapdata

Acceder.

```
ssh -L 1389:localhost:389 soffid01@172.16.9.20
```

```
ssh -L 1389:localhost:389 soffid02@172.16.9.35
```

Network Parameter	Authentication	Browser Options	Edit Options
Connection name: UNAL LDAP máquina 1			
Network Parameter			
Hostname:	localhost		▼
Port:	1389		▼

Crear dn raíz.

<div><div>▼</div><div>=</div><div>▼</div><div></div><div></div></div> <div><div>▼  DIT</div><div>▼  Root DSE (2)</div><div> o=unal.edu.co</div><div> Searches</div><div> Bookmarks</div></div>	<table><thead><tr><th>Attribute Description</th><th>Value</th></tr></thead><tbody><tr><td><b>objectClass</b></td><td><b>organization (structural)</b></td></tr><tr><td><b>objectClass</b></td><td><b>top (abstract)</b></td></tr><tr><td><b>o</b></td><td><b>unal.edu.co</b></td></tr></tbody></table>	Attribute Description	Value	<b>objectClass</b>	<b>organization (structural)</b>	<b>objectClass</b>	<b>top (abstract)</b>	<b>o</b>	<b>unal.edu.co</b>
Attribute Description	Value								
<b>objectClass</b>	<b>organization (structural)</b>								
<b>objectClass</b>	<b>top (abstract)</b>								
<b>o</b>	<b>unal.edu.co</b>								

Regenerar LDAP

```
docker stop soffid01-soffid-ldap-1
```

```
docker rm soffid01-soffid-ldap-1
```

```
docker volume rm -f compose_ldapdata compose_ldapconf
```



```
docker compose up -d
docker logs -f --since 5m soffid01-soffid-ldap-1

# Justo después de crearse falla, hay que iniciarla a mano
docker ps
docker start soffid01-soffid-ldap-1
```

## Carga de la configuración de LDAP de UNAL (ownCloud/soffid/proyectos/UNAL)

```
scp /home/svives/ownCloud/Soffid/Proyectos/UNAL/schema.sh soffid01@172.16.9.20:/home/soffid01/
scp /home/svives/ownCloud/Soffid/Proyectos/UNAL/import.sh soffid01@172.16.9.20:/home/soffid01/
scp /home/svives/ownCloud/Soffid/Proyectos/UNAL/ldapmcloud02-unal-2023-06-05-0318.ldif
soffid01@172.16.9.20:/home/soffid01/
chmod 777 schema.sh
chmod 777 import.sh
sudo vim import.sh ---> -H ldap://localhost/
docker compose cp schema.sh soffid-ldap:/tmp
docker compose cp import.sh soffid-ldap:/tmp
docker compose cp ldapmcloud02-unal-2023-06-05-0318.ldif soffid-ldap:/tmp
docker compose exec -it soffid-ldap bash
cd /tmp
./schema.sh
./import.sh
```

ldap\_sasl\_bind(SIMPLE): Can't contact LDAP server (-1) ---> revisar la cadena de conexión en el import.sh

```
root@026789f55d0c:/# cat import.sh
cat: import.sh: No such file or directory
root@026789f55d0c:/# cd /tmp
root@026789f55d0c:/tmp# ./import.sh
adding new entry "o=unal.edu.co"
ldap_add: Already exists (68)
```

```
soffid01@soffid01:~$ docker compose exec -it soffid-ldap bash
root@026789f55d0c:/# cd /tmp
root@026789f55d0c:/tmp# ./schema.sh
modifying entry "cn={0}core,cn=schema,cn=config"
ldap_modify: Insufficient access (50)
```

adding new entry "cn=Administracion Usuarios,o=unal.edu.co"

ldap\_add: Object class violation (65)

additional info: no structural object class provided

vim

```
:q! --> salir sin guardar cambios
:wq --> guardar y salir
/texto --> busca texto como en less
dd --> borrar la propia línea
dd3 --> borra las tres siguientes líneas
ctrl g --> te muestra la línea en la que estás
:1,17100 d --> borra las líneas desde la primera hasta la 17100
```

En la máquina 2 con datos finales:

```
scp /home/svives/ownCloud/Soffid/Proyectos/UNAL/schema.sh soffid02@172.16.9.35:/home/soffid02/
scp /home/svives/ownCloud/Soffid/Proyectos/UNAL/import.sh soffid02@172.16.9.35:/home/soffid02/
chmod 777 schema.sh
chmod 777 import.sh
cp /ldif/chingaza01-unal-2024-12-05-1421.ldif /home/soffid02/chingaza01-unal-2024-12-05-1421.ldif
```

```
sudo vim schema.sh
```

```
[1] ldapmodify \
[2] -D cn=SoffidAdmin,ou=builtin,dc=wfs,dc=aero \
[3] -w ThisPasswordIsSuperSecret \
[4] -H ldaps://cloud-adm02.soffid.com/ <<EOF
[5] =>
[6] ldapmodify -Y EXTERNAL -H ldapi:/// <<EOF
```

```
[7] borrada línea repetida:
```

```
[8] =>
```

```
[9] olcObjectClasses: {28}( 2.16.840.1.113730.3.2.4 NAME 'mailGroup' DESC 'UNAL'
```

```
[10] SUP top STRUCTURAL MUST cn MAY ( mail ) )
```

```
sudo vim import.sh
```

```
❏dapmcloud02-unal-2023-06-05-0318.ldif
```

```
❏=>
```

```
❏chingaza01-unal-2024-12-05-1421.ldif
```

```
❏-H ldaps://unal.demo.soffid.net/
```

```
❏=>
```

```
❏-H ldap://localhost/
```

```
❏fgrep -v passwordMaxFailure: |
```

```
❏=>
```

```
❏fgrep -v passwordMaxFailure: |
```

```
❏fgrep -v aci: |
```

```
docker compose cp schema.sh soffid-ldap:/tmp
```

```
docker compose cp import.sh soffid-ldap:/tmp
```

```
docker compose cp chingaza01-unal-2024-12-05-1421.ldif soffid-ldap:/tmp
```

```
docker compose exec -it soffid-ldap bash
```

```
cd /tmp
```

```
./schema.sh
```

```
./import.sh
```

```
----->>>
```

```
root@3fffd50e7f12:/tmp# ./import.sh
```

```
adding new entry "o=unal.edu.co"
```

```
ldap_add: Undefined attribute type (17)
```

```
❏additional info: aci: attribute type undefined
```

```
=====> añadir en import.sh --> fgrep -v aci: |
```

```
adding new entry "cn=Administracion Usuarios,o=unal.edu.co"
```

```
ldap_add: Object class violation (65)
```

```
❏additional info: no structural object class provided
```

```
====> objeto no cargado (:1,17941 d): # entry-id: 663 / dn: cn=Administracion Us arios,o=unal.edu.co
```

root@3fffd50e7f12:/tmp# ./import.sh

adding new entry "cn=Administración Mesa de Ayuda,o=unal.edu.co"

ldap\_add: Object class violation (65)

[additional info: no structural object class provided

]

==> objeto no cargado (:1,17 d): # entry-id: 664

adding new entry "cn=nsManagedDisabledRole,o=unal.edu.co"

ldap\_add: Object class violation (65)

[additional info: no structural object class provided

]

==> objeto no cargado (:1,118 d): # entry-id: 668 / dn: cn=nsManagedDisabledRole,o=unal.edu.co

adding new entry "cn=nsDisabledRole,o=unal.edu.co"

ldap\_add: Invalid syntax (21)

[additional info: objectClass: value #0 invalid per syntax

]

==> objeto no cargado (:1,17 d): # entry-id: 669 / dn: cn=nsDisabledRole,o=unal.edu.co

adding new entry "cn=nsAccountInactivationTmp,o=unal.edu.co"

ldap\_add: Invalid syntax (21)

[additional info: objectClass: value #1 invalid per syntax

]

==> objeto no cargado (:1,13 d): # entry-id: 670 / dn: cn=nsAccountInactivationTmp,o=unal.edu.co

[# entry-id: 671

[dn: cn=cn=nsDisabledRole\2Co=unal.edu.co,cn=nsAccountInactivationTmp,o=unal.ed

[ u.co

[nsUniqueld: e6ee777b-6b9711e6-b25c99d5-d316129f

[modifyTimestamp: 20160826141824Z

[createTimestamp: 20160826141824Z

[modifiersName: cn=directory manager

[creatorsName: cn=directory manager

[nsLicenseStartTime: 20040728192601Z

[cn: cn=nsDisabledRole,o=unal.edu.co

[cosPriority: 1

[nsAccountLock: true

[objectClass: top

```
[]objectClass: ldapsubentry
[]objectClass: extensibleobject
[]objectClass: costemplate
[]objectClass: nsLicenseUser
```

adding new entry "cn=nsAccountInactivation\_cos,o=unal.edu.co"

ldap\_add: Undefined attribute type (17)

[]additional info: cosAttribute: attribute type undefined

=====> añadir en import.sh --> fgrep -v cosAttribute: |

adding new entry "cn=nsAccountInactivation\_cos,o=unal.edu.co"

ldap\_add: Undefined attribute type (17)

[]additional info: cosspecifier: attribute type undefined

=====> añadir en import.sh --> fgrep -v cosspecifier: |

adding new entry "cn=nsAccountInactivation\_cos,o=unal.edu.co"

ldap\_add: Undefined attribute type (17)

[]additional info: costemplatedn: attribute type undefined

=====> añadir en import.sh --> fgrep -v costemplatedn: |

adding new entry "cn=nsAccountInactivation\_cos,o=unal.edu.co"

ldap\_add: Invalid syntax (21)

[]additional info: objectClass: value #1 invalid per syntax

====> objeto no cargado (:1,36 d): # entry-id: / 671 dn: cn=cn=nsDisabledRole\2Co=unal.edu.co...

[] # entry-id: 672 / dn: cn=nsAccountInactivation\_cos,o=unal.edu.co

adding new entry "uid=ajsanabriag,ou=People,o=unal.edu.co"

ldap\_add: Undefined attribute type (17)

[]additional info: mailAccessDomain: attribute type undefined

[]

=====> añadir en import.sh --> fgrep -v mailAccessDomain: []

====> objeto no cargado (:1,2958532 d): # entry-id: 87342 / dn: uid=eortizgo,ou=People,o=unal.edu.co

adding new entry "uid=userman04,ou=People,o=unal.edu.co"

ldap\_add: Invalid syntax (21)

□additional info: objectClass: value #4 invalid per syntax

==> objecto no cargado (:1,7658073 d): # entry-id: 294631 / dn: uid=userman04,ou=People,o=unal.edu.co

Ahora regeneramos la máquina 1:

```
# Desde la máquina 2 copiamos los ficheros a la 1

scp /home/soffid02/schema.sh soffid01@172.16.9.20:/home/soffid01/
scp /home/soffid02/import.sh soffid01@172.16.9.20:/home/soffid01/
scp /ldif/chingaza01-unal-2024-12-05-1421.ldif soffid01@172.16.9.20:/home/soffid01/

sudo vim chingaza01-unal-2024-12-05-1421.ldif
---> eliminamos entry-id: 663 664 668 669 670 671 672 87342 294631

docker compose cp schema.sh soffid-ldap:/tmp
docker compose cp import.sh soffid-ldap:/tmp
docker compose cp chingaza01-unal-2024-12-05-1421.ldif soffid-ldap:/tmp
docker compose exec -it soffid-ldap bash
cd /tmp
./schema.sh
./import.sh
```

soffidldap

Regenerar los syncservers

NOTA: el nombre de los parámetros hotname debe el nombre de la máquina

```
# Máquina 2

# Borrar el syncserver en la Consola
docker ps
docker stop soffid02-sync-server-1
docker rm soffid02-sync-server-1
docker volume rm -f compose_sync_conf
docker compose up -d
```

```
docker compose logs -f --since 5m sync-server
```

```
sudo vim /etc/hosts
```

```
172.16.9.20 soffid01
```

```
# Máquina 1
```

```
# Borrar el syncserver en la Consola
```

```
docker ps
```

```
docker stop soffid01-sync-server-1
```

```
docker rm soffid01-sync-server-1
```

```
docker volume rm -f compose_sync_conf
```

```
docker compose up -d
```

```
docker compose logs -f --since 5m sync-server
```

```
sudo vim /etc/hosts
```

```
172.16.9.35 soffid02
```

```
# Consola > Servidores de sincronización
```

```
-Xmx2048m
```

## Carga autoritativa tmp

Hacer una copia de la base de datos antes de la carga autoritativa por si acaso.

```
docker compose exec -it mariadb bash -c 'mariadb-dump -u root --password="dkF45.r4f" soffid' > mariadb-backup-20241205.sql
```

Borrar los datos actuales, por base de datos:

```
docker compose exec -it mariadb bash
```

```
mariadb -u root --password="$MYSQL_ROOT_PASSWORD"
```

user soffid;

select count(\*) from SC\_DADUSU;

select count(\*) from SC\_DADUSU where not (DUS\_IDUSU='99' or DUS\_IDUSU='1099' or DUS\_IDUSU='69702');

delete from SC\_DADUSU;

select count(\*) from SC\_USUARI;

select count(\*) from SC\_USUARI where not (USU\_CODI='admin' or USU\_CODI='svives' or USU\_CODI='gbuades');

delete from SC\_USUARI where not (USU\_CODI='admin' or USU\_CODI='svives' or USU\_CODI='gbuades');

---

Revision #36

Created 29 November 2024 18:16:50 by svives@soffid.com

Updated 8 January 2025 11:26:45 by svives@soffid.com