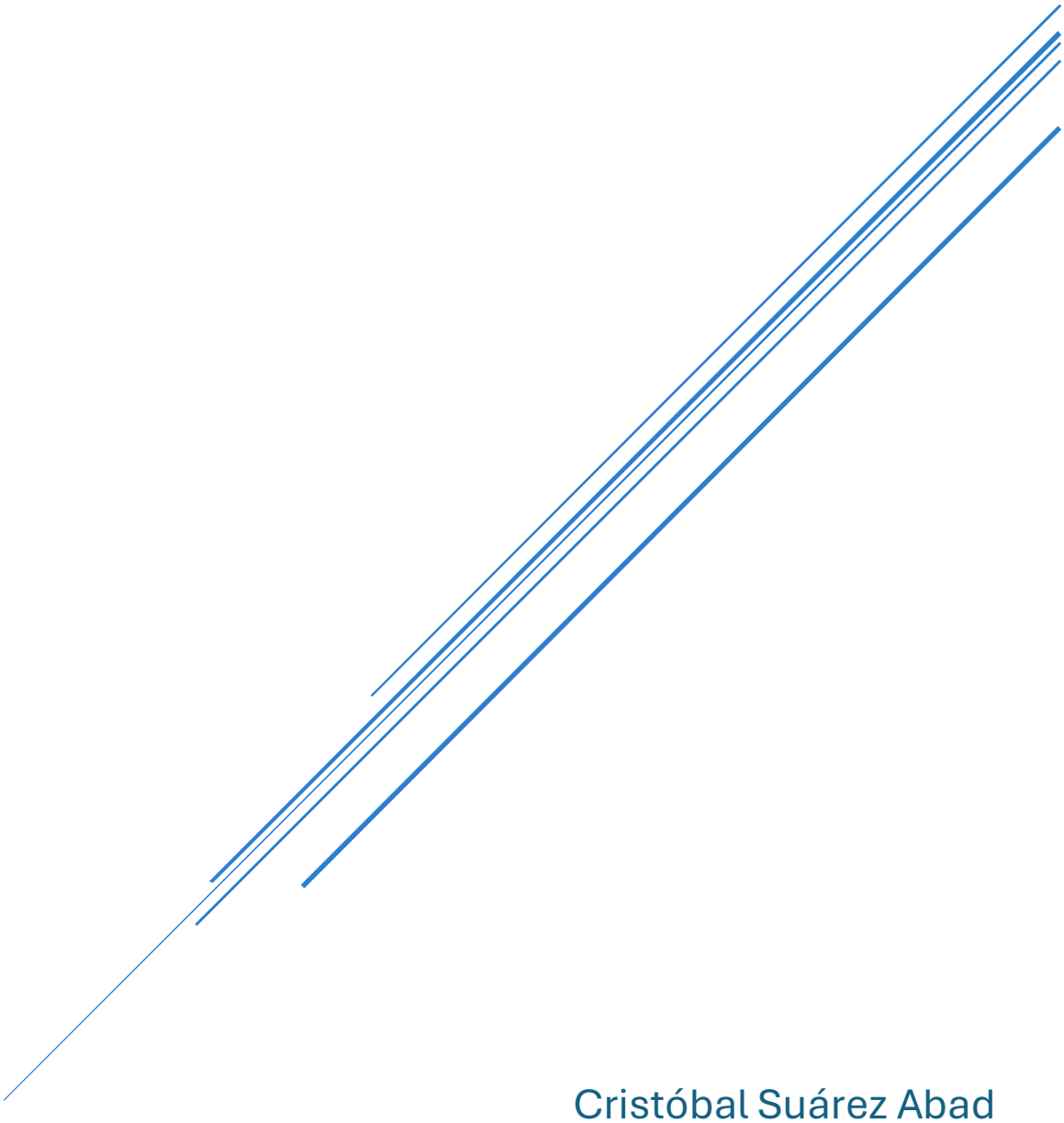
INSTALACIÓN DE CORREO CORPORATIVO CON POSTFIX Y DOVECOT.

Actividad 1.6



Cristóbal Suárez Abad
Administración de Sistemas Operativos - 2º ASIR

Contenido

Instalación de correo corporativo con Postfix y Dovecot.

1) Instalamos:

sudo apt update

sudo apt install postfix postfix-ldap dovecot-core dovecot-imapd dovecot-ldap libsasl2-modules-ldap

2) Configuración de archivos:

a) Creación de usuario para Dovecot:

Crearemos un usuario de sistema (vmail) sin privilegios de shell. Dovecot usará este usuario para escribir todos los correos en el disco, evitando problemas de permisos.

sudo useradd -r -s /sbin/nologin vmail

b) Configuración de Postfix:

sudo nano /etc/postfix/main.cf

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# See http://www.postfix.org/COMPATIBILITY_README.html -- default to 3.6 on
# fresh installs.
compatibility_level = 3.6

# --- Autenticación SASL vía Dovecot ---
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $mydomain


# --- Restricciones de Envío ---
smtpd_recipient_restrictions = permit_sasl_authenticated, permit_mynetworks,
reject_unauth_destination

# --- Configuración de Dominio y Entrega ---
virtual_mailbox_domains = SUAREZ1.abad2
virtual_transport = lmtp:unix:private/dovecot-lmtp
# --- Mapa de consulta LDAP ---
virtual_mailbox_maps = ldap:/etc/postfix/ldap-virtual-mailbox-maps.cf
```

```
# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_security_level=may
smtpd_use_tls = yes

smtp_tls_CApath=/etc/ssl/certs
smtp_tls_security_level=encrypt
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
smtp_use_tls = yes
# REPETIDA smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt


smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated
defer_unauth_destination
myhostname = control01.SUAREZ1.abad2
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, SUAREZ1.abad2, control01.SUAREZ1.abad2,
localhost.SUAREZ1.abad2, localhost
relayhost = [smtp.gmail.com]:587
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

sudo nano /etc/postfix/ldap-virtual-mailbox-maps.cf

server_host = 127.0.0.1
search_base = ou=usuarios,dc=SUAREZ1,dc=abad2
query_filter = (mail=%s)
result_attribute = mail
bind = yes
bind_dn = cn=admin,dc=SUAREZ1,dc=abad2
bind_pw = usuario.12345

sudo nano /etc/postfix/master.cf

```
#
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master" or
# on-line: http://www.postfix.org/master.5.html).
#
# Do not forget to execute "postfix reload" after editing this file.
#
#
==========================================================================
=
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (no)    (never) (100)
#
==========================================================================
=
smtp      inet  n       -       y       -       -       smtpd
#smtp      inet  n       -       y       -       1       postscreen
#smtpd     pass  -       -       y       -       -       smtpd
#dnsblog   unix  -       -       y       -       0       dnsblog
#tlsproxy  unix  -       -       y       -       0       tlsproxy
# Choose one: enable submission for loopback clients only, or for any client.
#127.0.0.1:submission inet n -   y       -       -       smtpd
submission inet n       -       y       -       -       smtpd
#  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_tls_auth_only=yes
#  -o smtpd_reject_unlisted_recipient=no
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#  -o smtpd_client_restrictions=$mua_client_restrictions
#  -o smtpd_helo_restrictions=$mua_helo_restrictions
#  -o smtpd_sender_restrictions=$mua_sender_restrictions
#  -o smtpd_recipient_restrictions=
#  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
#  -o milter_macro_daemon_name=ORIGINATING
# Choose one: enable smtps for loopback clients only, or for any client.
#127.0.0.1:smtps inet n -       y       -       -       smtpd
#smtps     inet  n       -       y       -       -       smtpd
#  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=no
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_reject_unlisted_recipient=no
```

```
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=
# -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
#628     inet n     -     y     -     -     qmqpd
pickup   unix n     -     y     60    1     pickup
cleanup  unix n     -     y     -     0     cleanup
qmgr     unix n     -     n     300   1     qmgr
#qmgr    unix n     -     n     300   1     oqmgr
tlsmgr   unix -     -     y     1000? 1     tlsmgr
rewrite  unix -     -     y     -     -     trivial-rewrite
bounce   unix -     -     y     -     0     bounce
defer    unix -     -     y     -     0     bounce
trace    unix -     -     y     -     0     bounce
verify   unix -     -     y     -     1     verify
flush    unix n     -     y     1000? 0     flush
proxymap unix -     -     n     -     -     proxymap
proxywrite unix -   -     n     -     1     proxymap
smtp     unix -     -     y     -     -     smtp
relay    unix -     -     y     -     -     smtp
     -o syslog_name=postfix/$service_name
#    -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq    unix n     -     y     -     -     showq
error    unix -     -     y     -     -     error
retry    unix -     -     y     -     -     error
discard  unix -     -     y     -     -     discard
local    unix -     n     n     -     -     local
virtual  unix -     n     n     -     -     virtual
lmtp     unix -     -     y     -     -     lmtp
anvil    unix -     -     y     -     1     anvil
scache   unix -     -     y     -     1     scache
postlog  unix-dgram n -    n     -     1     postlogd
#
# ====================================================================
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent.  See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# ====================================================================
#
```

```
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#
maildrop  unix -     n     n     -     -     pipe
  flags=DRXhu user=vmail argv=/usr/bin/maildrop -d ${recipient}
#
# ====================================================================
#
# Recent Cyrus versions can use the existing "lmtp" master.cf entry.
#
# Specify in cyrus.conf:
#  lmtp    cmd="lmtpd -a" listen="localhost:lmtp" proto=tcp4
#
# Specify in main.cf one or more of the following:
#  mailbox_transport = lmtp:inet:localhost
#  virtual_transport = lmtp:inet:localhost
#
# ====================================================================
#
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
#
#cyrus    unix -     n     n     -     -     pipe
#  flags=DRX user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension}
${user}
#
# ====================================================================
# Old example of delivery via Cyrus.
#
#old-cyrus unix -     n     n     -     -     pipe
#  flags=R user=cyrus argv=/cyrus/bin/deliver -e -m ${extension} ${user}
#
# ====================================================================
#
# See the Postfix UUCP_README file for configuration details.
#
uucp     unix -     n     n     -     -     pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
#
# Other external delivery methods.
#
ifmail   unix -     n     n     -     -     pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp    unix -     n     n     -     -     pipe
```

```
  flags=Fq. user=bsmtp argv=/usr/lib/bsmtp/bsmtp -t$nexthop -f$sender $recipient
scalemail-backend unix -      n       n       -       2       pipe
  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store ${nexthop}
${user} ${extension}
mailman   unix -      n       n       -       -       pipe
  flags=FRX user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py ${nexthop}
${user}
```

a) **Configuración de Dovecot.**

sudo nano /etc/dovecot/conf.d/10-mail.conf

```
mail_location = maildir:/var/mail/vhosts/%d/%n
mail_uid = vmail
mail_gid = vmail
```

sudo nano /etc/dovecot/conf.d/10-auth.conf

```
disable_plaintext_auth = no
auth_mechanisms = plain login
# <doc/wiki/UserDatabase.txt>

#!include auth-deny.conf.ext
#!include auth-master.conf.ext

!include auth-system.conf.ext
#!include auth-sql.conf.ext
!include auth-ldap.conf.ext
#!include auth-passwdfile.conf.ext
#!include auth-checkpassword.conf.ext
#!include auth-static.conf.ext
```

sudo nano /etc/dovecot/conf.d/10-master.conf

```
service lmtp {
 unix_listener /var/spool/postfix/private/dovecot-lmtp {
  mode = 0660
  user = postfix
  group = postdrop
 }

 # Create inet listener only if you can't use the above UNIX socket
 #inet_listener lmtp {
  # Avoid making LMTP visible for the entire internet
  #address =
  #port =
 #}
}

service auth {
 unix_listener /var/spool/postfix/private/auth {
  mode = 0660
```

```
  user = postfix
  group = postfix
 }

 # Postfix smtp-auth
 #unix_listener /var/spool/postfix/private/auth {
 #  mode = 0666
 #}

 # Auth process is run as this user.
 #user = $default_internal_user
}
```

sudo nano /etc/dovecot/conf.d/auth-ldap.conf.ext

```
# Authentication for LDAP users. Included from 10-auth.conf.
#
# <doc/wiki/AuthDatabase.LDAP.txt>

passdb {
 driver = ldap

 # Path for LDAP configuration file, see example-config/dovecot-ldap.conf.ext
 args = /etc/dovecot/dovecot-ldap.conf.ext
}

# "prefetch" user database means that the passdb already provided the
# needed information and there's no need to do a separate userdb lookup.
# <doc/wiki/UserDatabase.Prefetch.txt>
#userdb {
#  driver = prefetch
#}

userdb {
 driver = ldap
 args = /etc/dovecot/dovecot-ldap.conf.ext

 # Default fields can be used to specify defaults that LDAP may override
 #default_fields = home=/home/virtual/%u
}

# If you don't have any user-specific settings, you can avoid the userdb LDAP
# lookup by using userdb static instead of userdb ldap, for example:
# <doc/wiki/UserDatabase.Static.txt>
```

```
#userdb {
  #driver = static
  #args = uid=vmail gid=vmail home=/var/vmail/%u
#}
```

sudo nano /etc/dovecot/dovecot-ldap.conf.ext

```
# This file is commonly accessed via passdb {} or userdb {} section in
# conf.d/auth-ldap.conf.ext

# This file is opened as root, so it should be owned by root and mode 0600.
#
# http://wiki2.dovecot.org/AuthDatabase/LDAP
#
# NOTE: If you're not using authentication binds, you'll need to give
# dovecot-auth read access to userPassword field in the LDAP server.
# With OpenLDAP this is done by modifying /etc/ldap/slapd.conf. There should
# already be something like this:

# access to attribute=userPassword
#       by dn="<dovecot's dn>" read # add this
#       by anonymous auth
#       by self write
#       by * none

# Space separated list of LDAP hosts to use. host:port is allowed too.
hosts = 127.0.0.1

# LDAP URIs to use. You can use this instead of hosts list. Note that this
# setting isn't supported by all LDAP libraries.
#uris =

# Distinguished Name - the username used to login to the LDAP server.
# Leave it commented out to bind anonymously (useful with auth_bind=yes).
dn = cn=admin,dc=SUAREZ1,dc=abad2

# Password for LDAP server, if dn is specified.
dnpass = usuario.12345

# Use SASL binding instead of the simple binding. Note that this changes
# ldap_version automatically to be 3 if it's lower.
#sasl_bind = no
# SASL mechanism name to use.
#sasl_mech =
# SASL realm to use.
#sasl_realm =
# SASL authorization ID, ie. the dnpass is for this "master user", but the
# dn is still the logged in user. Normally you want to keep this empty.
#sasl_authz_id =
```

```
# Use TLS to connect to the LDAP server.
#tls = no
# TLS options, currently supported only with OpenLDAP:
#tls_ca_cert_file =
#tls_ca_cert_dir =
#tls_cipher_suite =
# TLS cert/key is used only if LDAP server requires a client certificate.
#tls_cert_file =
#tls_key_file =
# Valid values: never, hard, demand, allow, try
#tls_require_cert =

# Use the given ldaprc path.
#ldaprc_path =

# LDAP library debug level as specified by LDAP_DEBUG_* in ldap_log.h.
# -1 = everything. You may need to recompile OpenLDAP with debugging enabled
# to get enough output.
#debug_level = 0

# Use authentication binding for verifying password's validity. This works by
# logging into LDAP server using the username and password given by client.
# The pass_filter is used to find the DN for the user. Note that the pass_attrs
# is still used, only the password field is ignored in it. Before doing any
# search, the binding is switched back to the default DN.
auth_bind = yes

# If authentication binding is used, you can save one LDAP request per login
# if users' DN can be specified with a common template. The template can use
# the standard %variables (see user_filter). Note that you can't
# use any pass_attrs if you use this setting.
#
# If you use this setting, it's a good idea to use a different
# dovecot-ldap.conf.ext for userdb (it can even be a symlink, just as long as
# the filename is different in userdb's args). That way one connection is used
# only for LDAP binds and another connection is used for user lookups.
# Otherwise the binding is changed to the default DN before each user lookup.
#
# For example:
#   auth_bind_userdn = cn=%u,ou=people,o=org
#
#auth_bind_userdn =
```

```
# LDAP protocol version to use. Likely 2 or 3.
#ldap_version = 3

# LDAP base. %variables can be used here.
# For example: dc=mail, dc=example, dc=org
base = dc=SUAREZ1,dc=abad2

# Dereference: never, searching, finding, always
#deref = never

# Search scope: base, onelevel, subtree
scope = subtree

# User attributes are given in LDAP-name=dovecot-internal-name list. The
# internal names are:
#   uid - System UID
#   gid - System GID
#   home - Home directory
#   mail - Mail location
#
# There are also other special fields which can be returned, see
# http://wiki2.dovecot.org/UserDatabase/ExtraFields
#user_attrs = homeDirectory=home,uidNumber=uid,gidNumber=gid

# Filter for user lookup. Some variables can be used (see
# http://wiki2.dovecot.org/Variables for full list):
#   %u - username
#   %n - user part in user@domain, same as %u if there's no domain
#   %d - domain part in user@domain, empty if user there's no domain
user_filter = (&(objectClass=posixAccount)(uid=%u))

# Password checking attributes:
#  user: Virtual user name (user@domain), if you wish to change the
#       user-given username to something else
#  password: Password, may optionally start with {type}, eg. {crypt}
# There are also other special fields which can be returned, see
# http://wiki2.dovecot.org/PasswordDatabase/ExtraFields
#pass_attrs = uid=user,userPassword=password

# If you wish to avoid two LDAP lookups (passdb + userdb), you can use
# userdb prefetch instead of userdb ldap in dovecot.conf. In that case you'll
# also have to include user_attrs in pass_attrs field prefixed with "userdb_"
# string. For example:
#pass_attrs = uid=user,userPassword=password,\
```

```
#  homeDirectory=userdb_home,uidNumber=userdb_uid,gidNumber=userdb_gid

# Filter for password lookups
pass_filter = (&(objectClass=posixAccount)(uid=%u))

# Attributes and filter to get a list of all users
#iterate_attrs = uid=user
#iterate_filter = (objectClass=posixAccount)

# Default password scheme. "{scheme}" before password overrides this.
# List of supported schemes is in: http://wiki2.dovecot.org/Authentication
#default_pass_scheme = CRYPT

# By default all LDAP lookups are performed by the auth master process.
# If blocking=yes, auth worker processes are used to perform the lookups.
# Each auth worker process creates its own LDAP connection so this can
# increase parallelism. With blocking=no the auth master process can
# keep 8 requests pipelined for the LDAP connection, while with blocking=yes
# each connection has a maximum of 1 request running. For small systems the
# blocking=no is sufficient and uses less resources.
#blocking = no
```

3) Crear directorios de correo.

```
sudo mkdir -p /var/mail/vhosts/SUAREZ1.abad2
sudo chown -R vmail:vmail /var/mail/vhosts
sudo chmod -R 770 /var/mail/vhosts
```

4) Reiniciar servicios.

```
sudo systemctl restart postfix
sudo systemctl restart dovecot
sudo systemctl enable postfix dovecot
```

5) Creamos usuarios con correo en LDAP.

#MARTA

dn: uid=marta,ou=usuarios,dc=SUAREZ1,dc=abad2
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: marta
sn: suarez
uid: marta
uidNumber: 100389
gidNumber: 100389
homeDirectory: /home/marta
loginShell: /bin/bash
mail: marta@SUAREZ1.abad2
userPassword: usuario.12345

#MIGUEL

dn: uid=miguel,ou=usuarios,dc=SUAREZ1,dc=abad2
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: top
cn: miguel
sn: suarez
uid: miguel
uidNumber: 100390
gidNumber: 100390
homeDirectory: /home/miguel
loginShell: /bin/bash
mail: miguel@SUAREZ1.abad2
userPassword: usuario.12345

ldapadd -x -D "cn=admin,dc=SUAREZ1,dc=abad2" -w "usuario.12345" -f usuariocorreo.ldif

6) Instalamos swak para probar el correo.

sudo apt install swaks -y

Probamos el correo:

swaks --to miguel@SUAREZ1.abad2m --from marta@SUAREZ1.abad2 --server localhost:587 --auth LOGIN --auth-user marta -auth-password usuario.12345 --tls

Instalar OwnCloud

Vamos a usar un contenedor de Owncloud.

1) Instalamos Docker en el servidor.

Vamos a instalar en el servidor de LDAP un contenedor. Para ello debemos instalar previamente Docker.

Seguimos la guía de la página oficial: https://docs.docker.com/engine/install/ubuntu/

# 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 \
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

Luego procedemos con la instalación:

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

sudo systemctl status docker

Comprobamos: sudo docker run hello-world

Creamos el contenedor de owncloud:

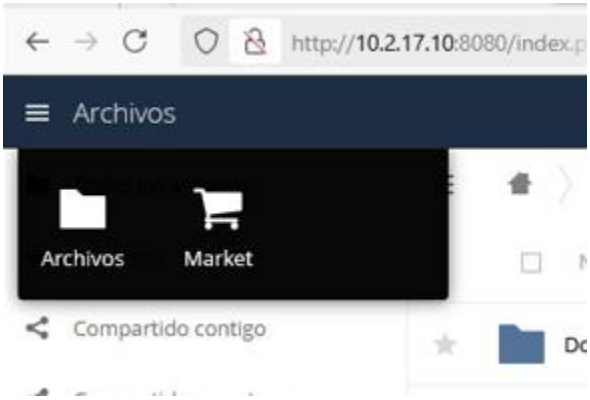
docker run -d --name ct_owncloud -p 8080:80 owncloud

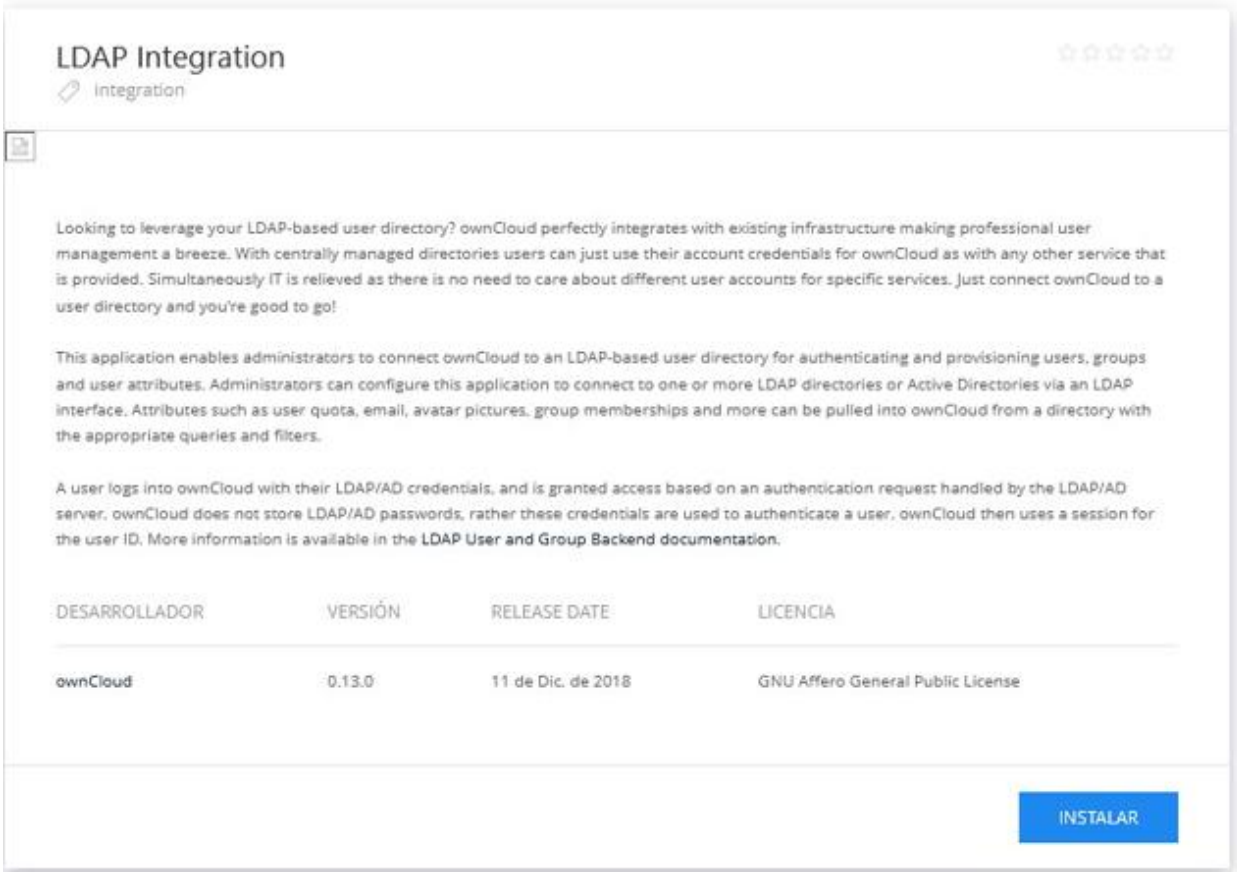
Una vez iniciado, ponemos en la URL de nuestro navegador web:

Ip_del_servidor:8080
Ejemplo: 10.2.17.10:8080

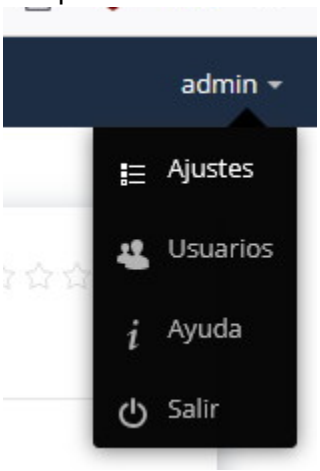
Después, para entrar usamos el usuario y la contraseña “admin”. Después establecemos un usuario principal (podemos usar el de antes).

Una vez dentro vamos a Market e instalamos el modulo de LDAP Integration.

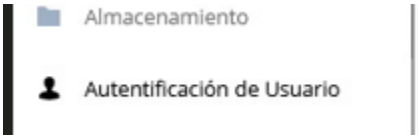




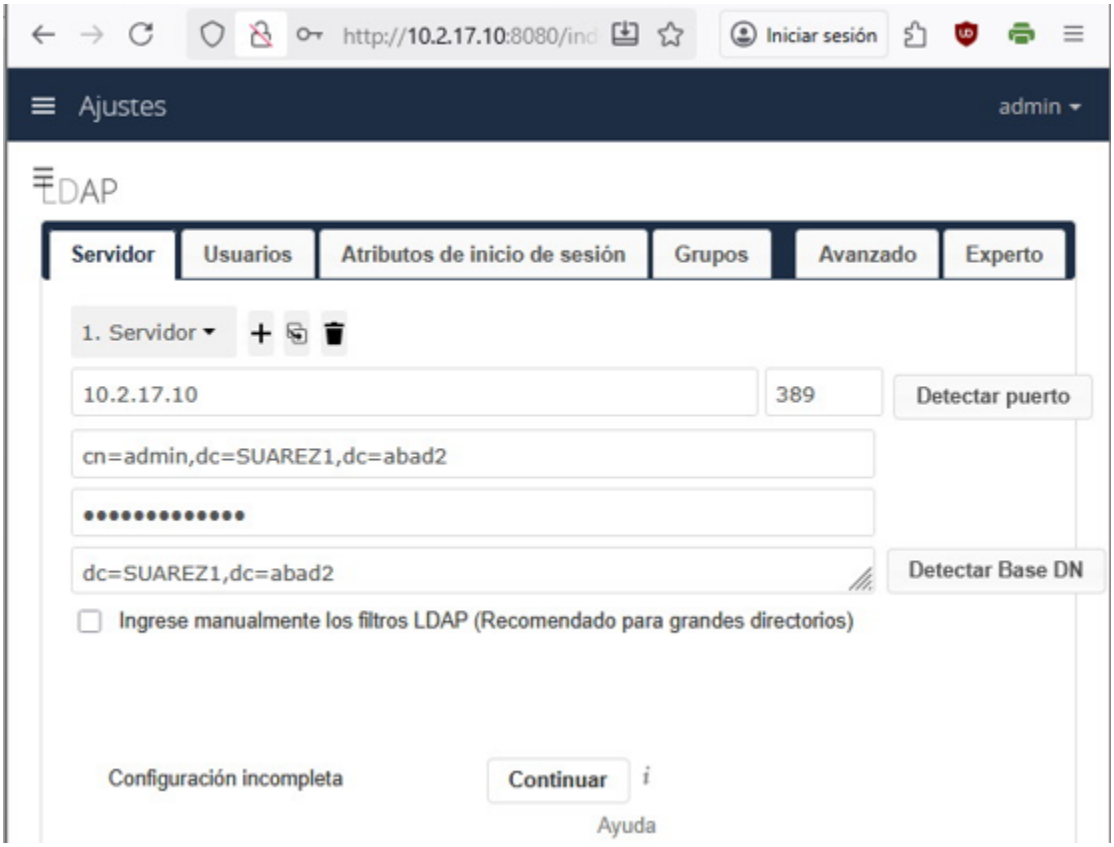
Después nos vamos a “Ajustes”.

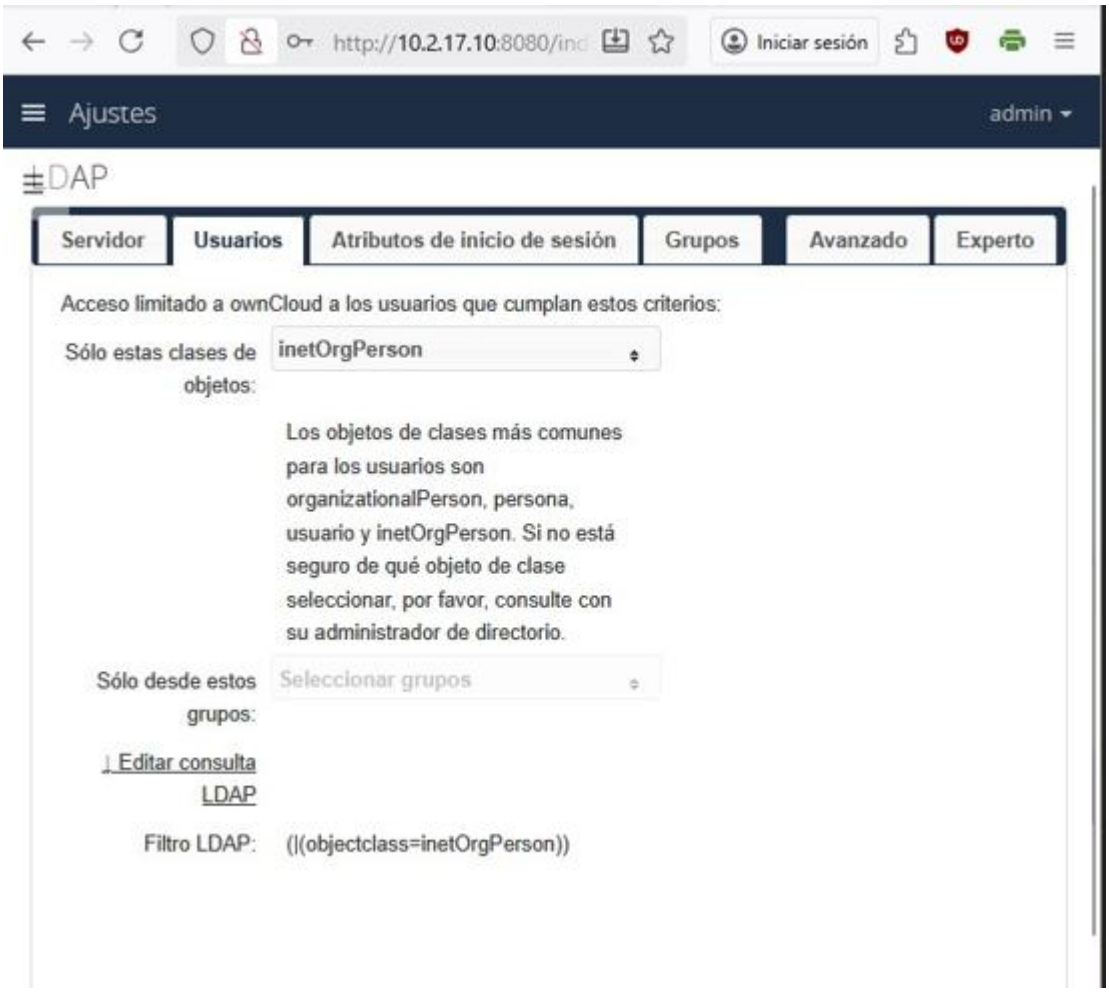


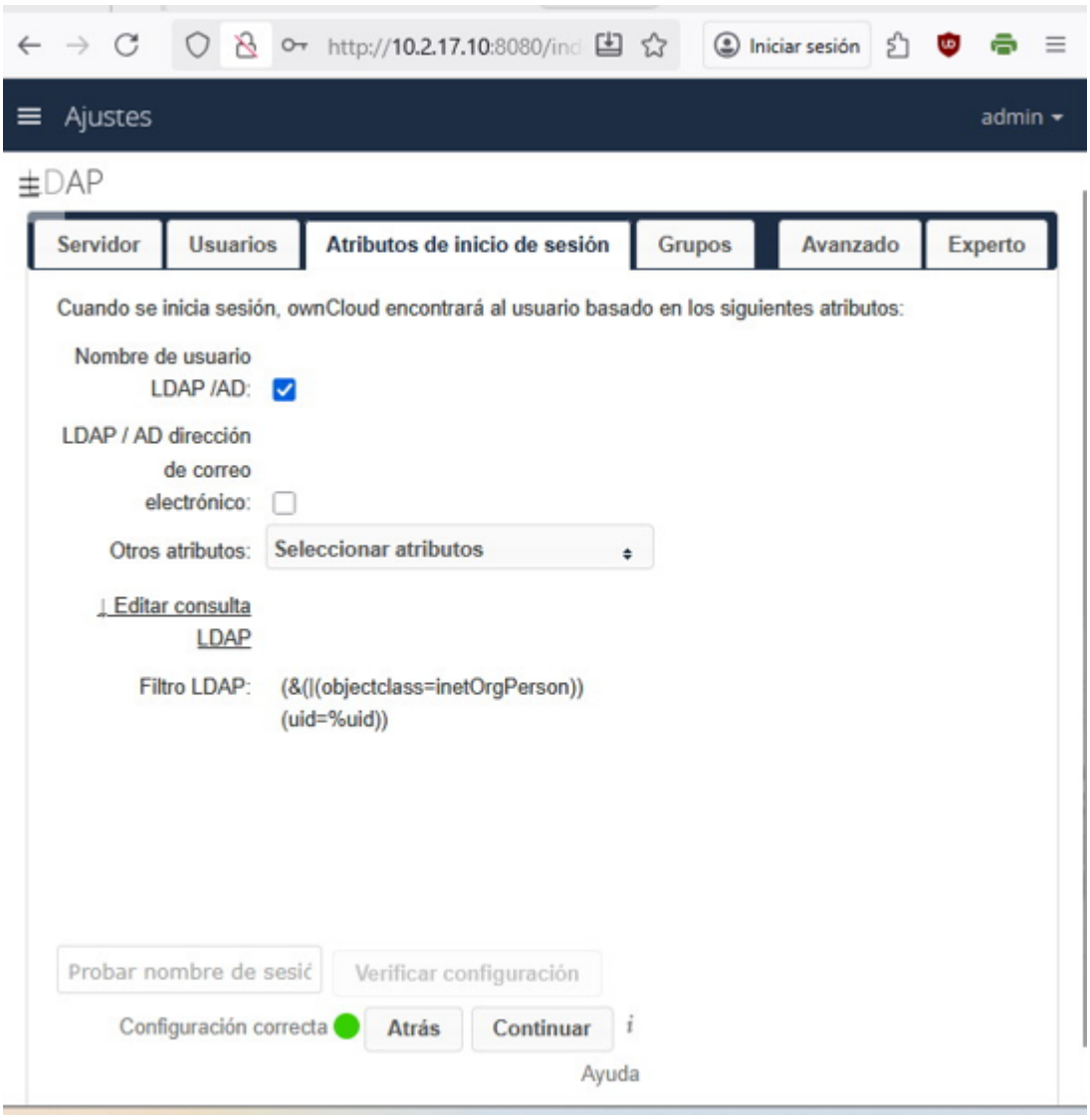
Autenticación de Usuario.

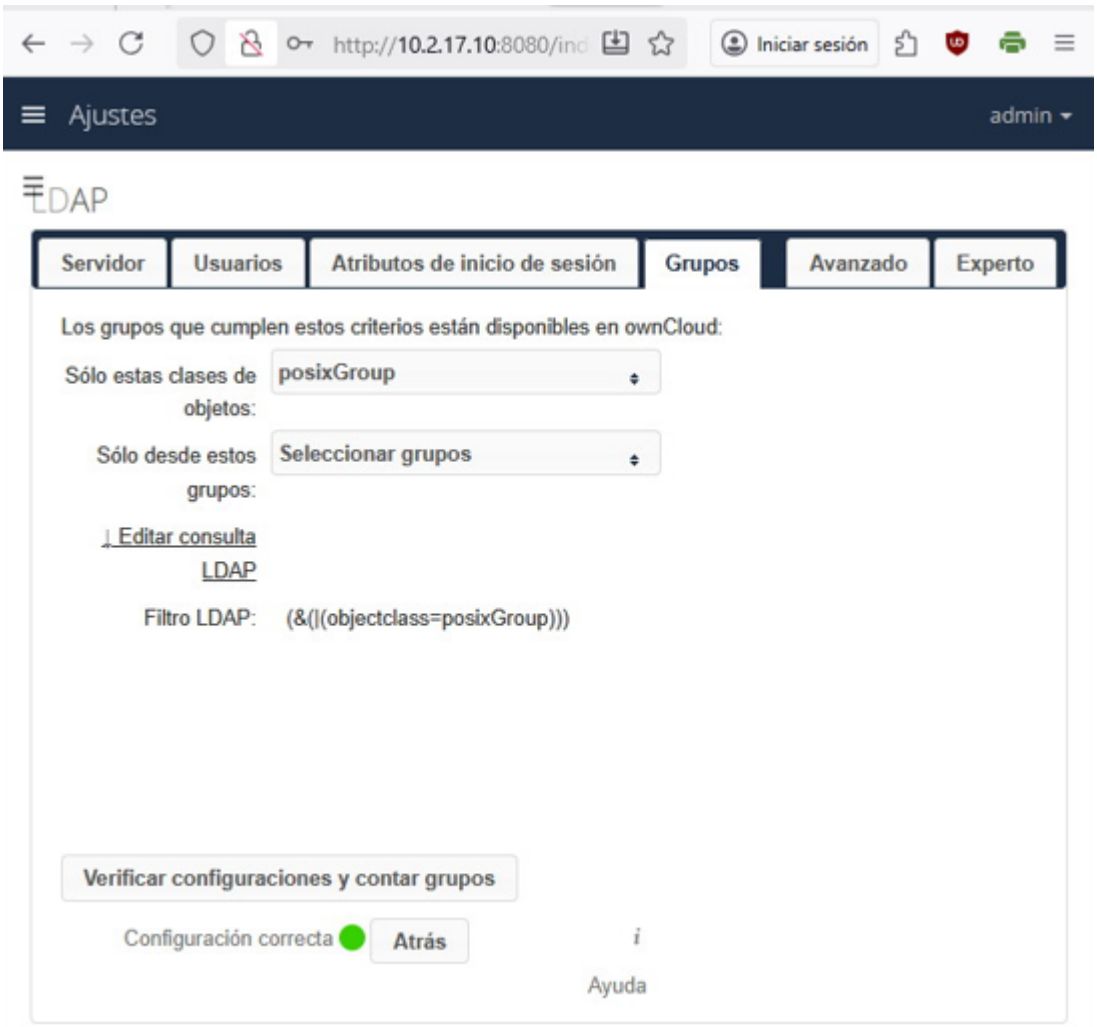


Rellenamos la información con los datos de nuestro servidor.

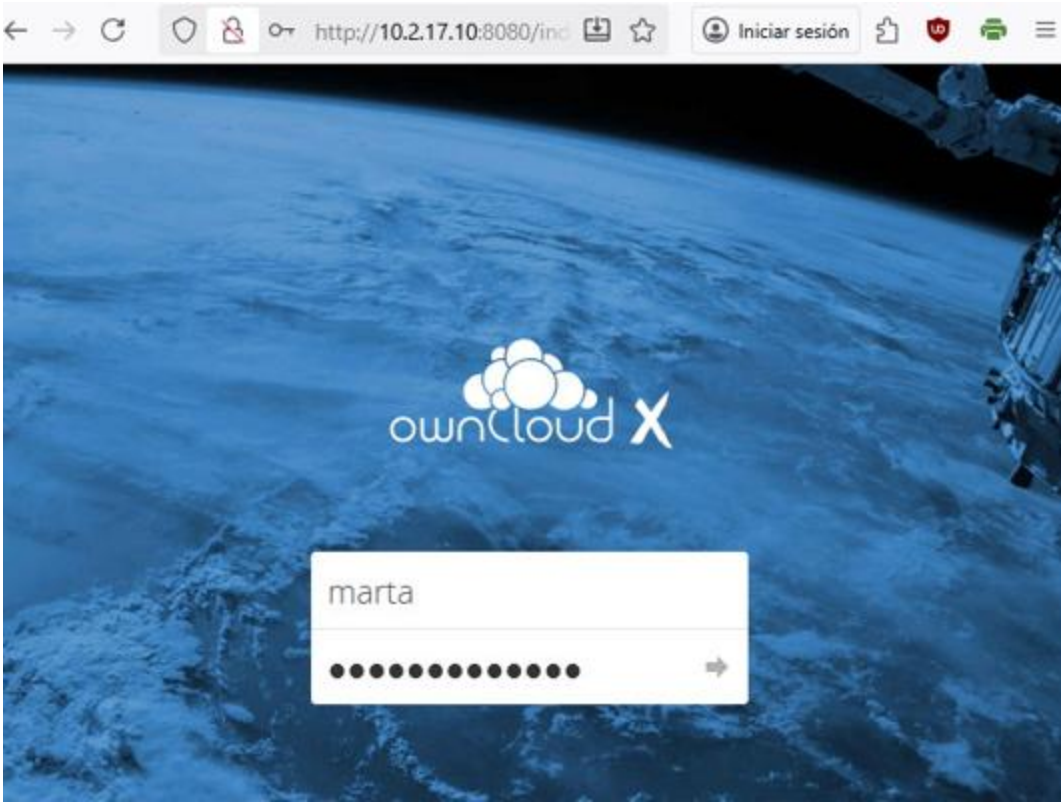



http://10.2.17.10:8080/ind
Iniciar sesión
Ajustes
admin
LDAP
Servidor
Usuarios
Atributos de inicio de sesión
Grupos
Avanzado
Experto
Acceso limitado a ownCloud a los usuarios que cumplan estos criterios:
Sólo estas clases de objetos:
inetOrgPerson
Los objetos de clases más comunes para los usuarios son organizationalPerson, persona, usuario y inetOrgPerson. Si no está seguro de qué objeto de clase seleccionar, por favor, consulte con su administrador de directorio.
Sólo desde estos grupos:
Seleccionar grupos
Editar consulta LDAP
Filtro LDAP:
(|(objectclass=inetOrgPerson))

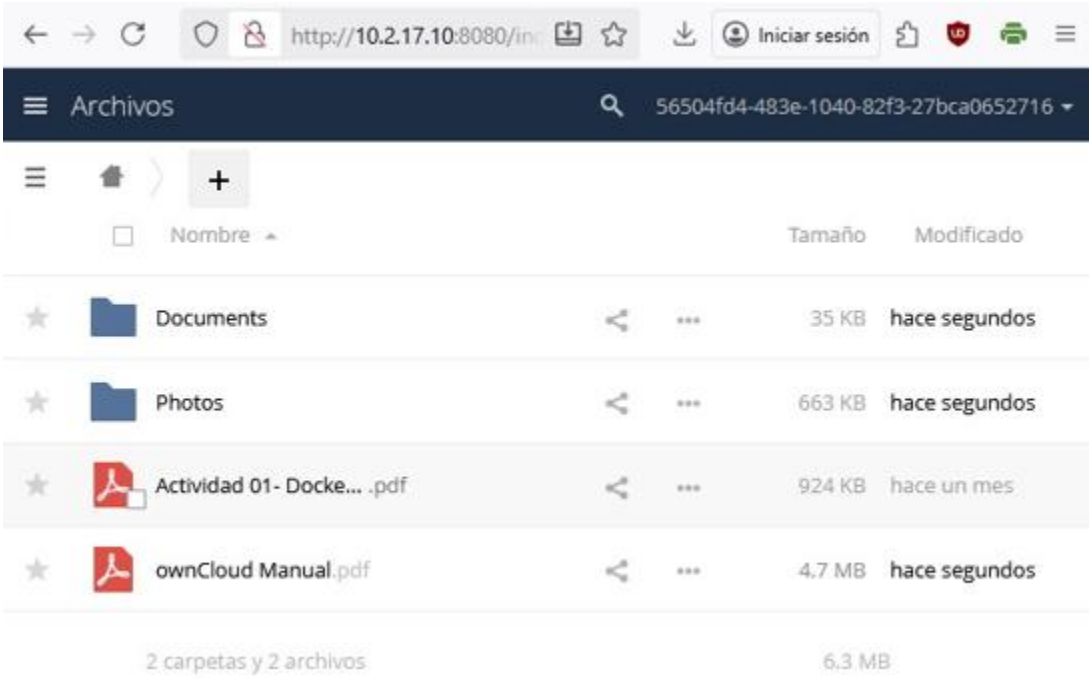

http://10.2.17.10:8080/inc
Iniciar sesión
Ajustes
admin
LDAP
Servidor
Usuarios
Atributos de inicio de sesión
Grupos
Avanzado
Experto
Cuando se inicia sesión, ownCloud encontrará al usuario basado en los siguientes atributos:
Nombre de usuario LDAP /AD:
LDAP / AD dirección de correo electrónico:
Otros atributos:
Seleccionar atributos
Editar consulta LDAP
Filtro LDAP:
(&(|(objectclass=inetOrgPerson))
(uid=%uid))
Probar nombre de sesić
Verificar configuración
Configuración correcta
Atrás
Continuar
Ayuda



Después salimos y comprobamos su funcionamiento con otro usuario.

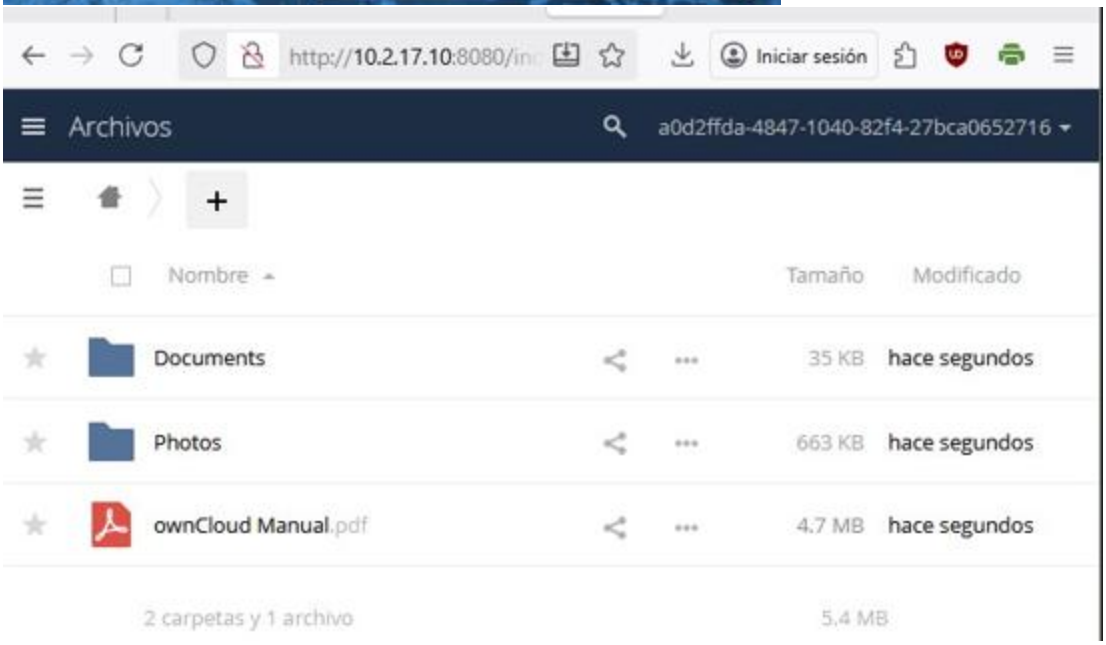


Guardamos un documento en el usuario Marta.



Nos salimos y comprobamos con otro usuario que solo está para Marta.


http://10.2.17.10:80
ownCloud X
miguel


http://10.2.17.10:8080/in
Iniciar sesión
Archivos
a0d2ffda-4847-1040-82f4-27bca0652716
Nombre
Tamaño
Modificado
Documents
35 KB
hace segundos
Photos
663 KB
hace segundos
ownCloud Manual.pdf
4.7 MB
hace segundos
2 carpetas y 1 archivo
5.4 MB