

Contents

 1 PHP frissítése
 2 Telepítés és beállítások
◆ 2.1 Install
 2.2 MvSQL adatbázis
 2.3 Data mappa létrehozása
♦ 2 4 Config php
▲ 2.5 Anache beállítások
0 2 5 1 Virtuális hoszt
Δ 2.5.1 VIIIUUIIS 10321
V 2.3.2 11313 ▲ 2.6 Ele2 indítée
♦ 2.0 EIS (IIIUIIdS) ♦ 2.7 Crop ich hoállítána
• 3 In-Memory cache
♦ 3.2 APGU
Ø 3.2.1 Apcu cache megjelenit
 4 Tovabbi beallitasok
4.1 Performancia tuning
 4.2 Kiegészít?k
♦ 4.3 Vírus sz?rés
♦ 4.3.1 SELinux
◊ 4.3.2 Tesztelés
 4.4 OCC konzol
 5 Felhasználói kézikönyv
♦ 5.1 Klinesek
♦ 5.1.1 Linux kliens
♦ 5.1.2 Windows kliens
♦ 5 1 3 Android
0.5.1.4 IPhone
6 Trouble shooting

6.1 Nem lehet megnyitni az Admin fület

PHP frissítése

Az OwnCloud 9.x használatához legalább PHP 5.6 ra van szükség. Ehhez fel kell telepíteni a "remi" repozitorikat:

```
wget http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
rpm -Uvh remi-release-6*.rpm epel-release-6*.rpm
```

Kapcsoljuk be az alap "remi" repo használatát valamint a php56 repot: /etc/yum.repos.d/remi.repo

```
[remi]
name=Les RPM de remi pour Enterprise Linux 6 - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/6/remi/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/6/remi/$basearch/
mirrorlist=http://etc/pki/rpm-gpg/RPM-GPG-KEY-remi
...
[remi-php56]
name=Les RPM de remi de PHP 5.6 pour Enterprise Linux 6 - $basearch
#baseurl=http://rpms.famillecollet.com/enterprise/6/php56/$basearch/
mirrorlist=http://rpms.famillecollet.com/enterprise/6/php56/mirror
# WARNING: If you enable this repository, you must also enable "remi"
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
Majd frissítsük a PHP-t.
```

yum update php*

Ellen?rizzük az php verziót:

php -v PHP 5.6.26 (cli) (built: Sep 15 2016 14:57:05) Copyright (c) 1997-2016 The PHP Group Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies

Telepítés és beállítások

```
Install
```

Az OwnCloud biztosít yum repozitorit a telepítéshez. Felteszi a szükséges php csomagokat, valamit a helyére másolja a http fájlokat: /var/www/html/owncloud

ownCloud repos: http://download.owncloud.org/download/repositories/9.1/owncloud/ manual: https://doc.owncloud.org/server/9.1/admin_manual/installation/linux_installation.html#changes-in-owncloud-9

Több repozitory is elérhet?.

- CentOS_6_SCL_PHP54 owncloud-9.1.1-1 -> cseljes csomag php 5.4-el
 CentOS_6_SCL_PHP54 owncloud-files-9.1.1-1 -> csak a forrás PHP 5.4-hez
 CentOS_6 owncloud-9.1.1-1 -> teljes csomag > PHP 5.4 [ezt fogjuk mi is használni]. Ez tartalmazza a owncloud-files-9.1.1-1 csomagot is.
- CentOS_6 owncloud-files-9.1.1-1 -> csak a forrás, a függ?ségek nem. Részhalmaza a owncloud-9.1.1-1 csomagnak.

Telepítsük a repozitoryt:

rpm --import https://download.owncloud.org/download/repositories/9.1/CentOS_6/repodata/repomd.xml.key wget http://download.owncloud.org/download/repositories/9.1/CentOS_6/ce:9.1.repo -0 /etc/yum.repos.d/ce:9.1.repo # yum clean expire-cache

Majd az összes szükséges fájlt telepíthetjük yum-al:

yum install owncloud

A telepítés után létrejött a /var/www/html/ownclooud mappa. Ehhez létre fogunk hozni egy apache virtuális hoszt fált.

MySQL adatbázis

A közösségi verziója az ownCloud-nak támogatja a MySQL/MariaDB adatbázist is, ezt fogjuk használni. Hozzunk létre egy új MySQL felhasználót az ownCloud-nak. Ezt a felhasználót az összes joggal fel kell ruházni, nem elég csak a telepítési idejére beállítani a DDL jogokát, azokra kés?bb is szüksége lesz, pl, a pluginek be-ki kapcsolása közben is módosítja az adatbázis struktúrát. Ezen felül fontos, hogy legyen "LOCK TABLES" joga is, mert mikor lefut majd (negyed óránként) a cron.php, lock-olni fog bizonyos táblákat.

Az adatbázis struktúrát majd az els? indításkor létre tudjuk hozni a böngész?ben futó Wizard folyamat segítségével.

Data mappa létrehozása

Biztonsági megfontolásból fontos, hogy a data mappa, ahol a felhasználók fájljait tárolja az ownCloud kívül legyen a /var/www mappában ugyanakkor rendelkezzen a megfelel? SELinux kontextust. A data mappat a /var/owncloud mappaba fogjuk tenni:

```
mkdir /var/owncloud
# chown root.apache /var/owncloud/
```

Másoljuk át a jelenlegi data mappa tartalmát. Fontos hogy használjuk a -p kapcsolót, ami hatására meg?rzi a jogokat, tulajdonost:

```
# cp -rp /var/www/html/ownciou
# chmod 700 /var/owncloud/data
            /var/www/html/owncloud/data/ /var/owncloud/data
```

Végezetül állítsuk be az SELinux kontextust, különben az apache nem fog hozzáférni:

```
# semanage fcontext -a -t httpd_sys_rw_content_t '/var/owncloud/data'
# restorecon '/var/owncloud/data'
```

Config.php

A /var/www/html/owncloud/config/config.php-ben be kell állítani az adatbázist, a data mappa helyét.

```
<?php
>rpip
$CONFIG = array (
   'updatechecker' => false,
   'instanceid' => 'ocg8blblbuhs',
   'passwordsalt' => 'nqdeRpx3jXP9R+7QA5Uh9wZhEkargm',
   'secret' => 'dDPRFMidUmgbTQCPZjMe79vjohVem9rIJC7GoZlrJF5dK/Fz',
   'trusted_domains' =>
           array (
    0 => 'owncloud.berki2.org',
         0 => 'owncloud.berki2.org',
),
'datadirectory' => '/var/owncloud/data',
'overwrite.cli.url' => 'http://owncloud.berki2.org',
'dbtype' => 'mysql',
'version' => '9 1.1.3',
'dbname' => 'owncloud',
'dbhost' => 'localhost',
'dbuser' => 'owncloud',
'dbuser' => 'owncloud',
'dbuser' => 'owncloud',
'dbpassword' => 'liteon',
'logtimezone' => 'UTC',
'installed' => true,
'mail_smtpmode' => 'php',
'mail_from_address' => 'owncloud',
'mail_from_address' => 'localhost',
'mail_smtphost' => 'localhost',
'mail_domain' => 'berki2.org',
'enable_avatars' => true,
'logfile' => '/var/log/owncloud.log',
'loglevel' => '3';
 );
```

Állítsuk be a data mappa helyét, amit fent hoztunk létre, adjuk meg az adatbázis elérhet?ségeket valamint az email szerver beállításait.

Apache beállítások

Virtuális hoszt

/etc/httpd/vhosts.d/owncloud.berki.org.80.conf

ServerAlias owncloud.berki2.org ErrorLog logs/owncloud.berki2.org-error_log CustomLog logs/owncloud.berki2.org-access_log common

Redirect permanent / https://owncloud.berki2.org

</VirtualHost>

/etc/httpd/vhosts.d/owncloud.berki.org.443.conf

```
<VirtualHost *:443>
         ServerAdmin adam@proarcok.com
ServerName owncloud.berki2.org:443
         ServerAlias owncloud.berki2.org:443
         ErrorLog logs/owncloud.berki2.org-error_log
         CustomLog logs/owncloud.berki2.org-access_log common
```

##Fel evre garantaljuk, hogy HTTPS-en maradunk Headersetw&ysict-Transport-Security "max-age=15768000; includeSubdomains;"

```
SSL parameterek
SSLEngine on
```

- # Here, I am allowing only "high" and "medium" security key lengths. SSLCipherSuite HIGH:MEDIUM
- # Here I am allowing SSLv3 and TLSv1, I am NOT allowing the old SSLv2. SSLProtocol all -SSLv2

- # Server Certificate:
 - SSLCertificateFile /etc/httpd/ssl/owncloud.berki2.org-server.cer
- # Server Private Key: SSLCertificateKeyFile /etc/httpd/ssl/owncloud.berki2.org-server.key

Documentapowww/html/owncloud/"

#Alias /owncloud "/var/www/html/owncloud/" <Direcaprwww/html/owncloud"> Options +FollowSymLinks

```
AllowOverride All
<IfModule mod dav.c>
```

```
Dav off
```

</IfModule>

```
SetEnv HOME /var/www/html/owncloud
         SetEnv HTTP_HOME /var/www/html/owncloud
</Directory>
<Directorywww/html/owncloud/data/">
# just in case if .htaccess gets disabled
Require all denied
</Directory>
```

</VirtualHost>

HSTS

(HTTP Strict Transport Security)

A HSTS egy biztonsági kiegészítése a https protokollnak, amit az RFC 6797 ír le: (https://tools.ietf.org/html/rfc6797). Ha a HSTS az apache szerveren be van kapcsolva egy HTTPS alatt kommunikáló virtuális hoszton, akkor a szerver a válaszba elhelyez egy HTST header-t is, amiben egy vállalást tesz arra, hogy legalább az itt megadott ideig (általában fél év) az adott hosztot kizárólag HTTPS alatt fogja futtatni. Ha a kliens (böngész? vagy más mobilos alkalmazás) HSTS képes, akkor el fogja menteni ezt az információt az adott hoszt névhez. Ha bármilyen módon megpróbálná kés?bb egy támadó a HTTPS kommunikációt HTTP-re terelni (akár egy hamis oldalon keresztül) a böngész? nem lesz hajlandó kommunikálni HTTP felett amíg a vállalás le nem jár. Sikeres kapcsolat esetén a vállalás ideje mindig úraindul nem jár. Sikeres kapcsolat esetén a vállalás ideje mindig újraindul.

A httpd.conf-ban töltsük be a htst modult. /etc/httpd/conf/httpd.conf

LoadModule headers_module modules/mod_headers.so

Ezután a HTTPS-es (443) virtuális hoszt konfigurációjába be kell kapcsolni a HSTS header használatát, és meg kell adni a vállalási id?t is. /etc/httpd/vhosts.d/owncloud.berki.org.443.conf

. . Header always set Strict-Transport-Security "max-age=15768000; includeSubdomains;" . . .

A max-age paramétert másodpercben kell megadni, mi fél évet állítunk be.

Indítsuk újra az apache-ot.

service httpd restart

Majd nyissuk meg megint a https://owncloud.berki2.org oldalt. Nézzük meg a fogadott fejléceket a Firefox fejleszt?i eszközben.



A Firefox a HSTS információkat a iteSecurityServiceState.txt fájlban tárolja:

```
$ cat /home/adam/.mozilla/firefox/61og30yv.default/SiteSecurityServiceState.txt | grep berki2
owncloud.berki2.org:HSTS 0 17089 1492295509620,1,1
```

Els? indítás

Miután készen van a virtuális hosz, els? indításkor elvégezhetjük a képerny?n keresztül a rendszer alap beállításait. Navigáljunk a https://owncloud.berki2.org címre.

...

Cron job beállítása

Állítsjuk be, hogy az owncloud system cront-t használjon:



Majd hozzuk létre a cron fájlt /etc/cron.d/owncloud

*/15 * * * * apache /usr/bin/php -f /var/www/html/owncloud/cron.php

És adjunk rá megfelel? jogot:

chmod 644 ownlcoud



Nagyon fontos, hogy az owncloud adatbázis felhasználónak legyen LOCK TABLE joga, különben nem fog lefutni a cron.php script

in-Memory cache

Memória cache használata nélkül az openCloud borzasztóan lassú. Háromféle memória cache megoldást támogat az ownCloud.

Redis

Mi a Redis in-memory adatstruktúra adatbázist fogjuk használni, melynek telepítését és finomhangolását itt találjuk: Redis -NoSQL database

Ha a Redis szerverünk a localhost:6379 -on fut, akkor az alábbi sorokat kell elhelyezni a config.php-ben hogy a redis szervert használja az owncloud:

```
'redis' => array(
    'host' => 'localhost',
    'port' => 6379,
    ),
```

A SELinux-ban az apache-nak meg kell engedni, hogy hálózati kapcsolatokat hozzon létre, hogy tudjon kapcsolódni a redis szerverhez.

setsebool -P httpd_can_network_connect 1

APCu

yum install php-pecl-apcu

https://anavarre.net/how-to-monitor-and-tune-apcu/

/etc/php-zts.d/40-apcu.ini

bekapcsolja a cache-t
apc.enabled=1

#hány memória caches legyen. (Akárcsak a redis-nél, itt is több különálló cache lehet)
apc.shm_segments=1

#memória mérete
apc.shm_size=5G

#fájméret amit cache-be rak
apc.max_file_size=1M

apc.stat=1

#Maximum number of files APC can store (rotation).
apc.num_files_hint=200000

#Maximum number of users data entries that APC can store. apc.user_entries_hint=20000

Majd az owncloud config.php-be az alábbi sort kell elhelyezni:

'memcache.local' => '\OC\Memcache\APC',

Apcu cache megjelenít?

Töltsük le innen az apcu megjelenít? scrpt-et: https://raw.githubusercontent.com/krakjoe/apcu/master/apc.php

Tegyük a webszerver mappájába. Adjunk rá végrehajtási jogot. Írjuk át a fájl elején a felhasználó név és jelszó párost:

defaults('ADMIN_USERNAME','user');
defaults('ADMIN_PASSWORD','new-password');

Majd nyissuk meg: https://<path>/apc.php



További beállítások

Performancia tuning

Apaceh beállítások

MaxKeepAliveRequests 10000 ... <IfModule prefork.c> StartServers 8 MinSpareServers 5 MaxSpareServers 20 ServerLimit 256 MaxClients 256 MaxRequestsPerChild 4000 </IfModule>

Kiegészít?k

Vírus sz?rés

Kapcsoljuk be a " Antivirus App for files " kiegészít?t.



Majd konfiguráljuk. A ClamAV-t a socket-en keresztül kell hívja: /var/run/clamav/clamd.sock

Antivirus Configuration		
Mode	Daemon (Socket) 🔻	
Socket	/var/run/clamav/clam	
Stream Length	1024	bytes
Action for infected files found while scanning	Delete file 🔻	
Save		

SELinux

Ahhoz hogy a HTTP (owncloud) hozzáférjen a ClamAV sockethez (/var/run/clamav/clamd.sock) szükség van egy új SELinux szabályra. Hozzuk létre a **owncloudSELinux.te** fájlt.

Majd fordítsuk le és töltsük be:

checkmodule -M -m -o owncloudSELinux.mod owncloudSELinux.te # semodule_package -o owncloudSELinux.pp -m owncloudSELinux.mod # semodule -i owncloudSELinux.pp

Tesztelés

Töltsünk le innen egy teszt vírust:

A webes felületen keresztül töltsük fel. A képerny?n megjelenik a figyelmeztetés:

Error uploading file "eicar.com.txt": Virus Eicar-Test-Signature is detected in the file. Upload cannot be completed. 🗙

A /var/log/messages fájlban láthatjuk hogy a ClamAV vírusnak titulálta a fájlt:

Oct 15 13:38:08 centostest clamd[1182]: instream(local): Eicar-Test-Signature FOUND

OCC konzol

Az ownCloud-nak van egy parancssori konzola, ahol az összes admin funkció elvégezhet?

Felhasználói kézikönyv

Klinesek

Linux kliens

Els?ként az certificate authority-t be kell importálni a linux-ba mint biztonságos kulcs:

- # cp berki-ca.pem /etc/pki/ca-trust/source/anchors/
 # update-ca-trust extract
- A kliens tekintetében két lehet?ségünk van. Használhatjuk a gnome-ba beéített OwnCloud klienst.



A Nautilus-ban fog csak megjelenni a megosztás, Krusader-ben nem fog látszani. De a legnagyobb baj vele, hogy valós id?ben szinkronizál a szerverrel, így bármit módosítunk, meg kell várni, hogy felküldje a szerverre, ezért borzasztó lassú.

Ennél sokkal jobb az OwnCloud oldaláról letölthet? kliens, ami okosan, a háttérben szinkronizál egy megadott mappába, amit így a Krusader-ben is láthatunk. Rengeteg finom beállítási lehet?ség van benne.

ownCloud	×
adam ovyriclou_rklorg	
Connected to https://owncloud.berki.org as adam.	Account
 w workloud Synchronizing with local folder ownCloud w w w box w w w box Photos (503 MB) Add Folder Sync Connection 	
	X Close

Az owncloud biztosít egy dnf repót. Fedora 24-hez az alábbit kell használni:

dnf config-manager --add-repo http://download.opensuse.org/repositories/isv:ownCloud:desktop/Fedora_24/isv:ownCloud:desktop.repo dnf install owncloud-client

Windows kliens

Android

Els?ként az andorid fizet?s klienst telepítsük föl az Appstore-ból



A link a webes ownlcoud felületen a bal fels? sarokban a Personal menüpontban elérhet?

Sajnos a fájlok tárolási helye nem megváltoztatható, mindent a bels? memóriában fog tárolni. Ezért át kell helyezzük a teljes adat mappáját a bels? memóriából az SD kártyára, majd egy szimbolikus linket létrehozni a régi helyre. Ehhez remek Android-os eszközök léteznek, mi a **FolderMount** alkalmazás ingyenes változatát fogjuk használni: https://play.google.com/store/apps/details?id=com.devasque.fmount&hl=en



De el?tte root-olni kell a telefont. Ezt a KingRoot programmal fogjuk megtenni.



A mobil böngész?ben navigáljunk a https://kingroot.net/ oldalra.



Kizárólag innen töltsük le a KingRoot-ot, ha máshonnan töltjük le az vírusos lehet

Majd telepítsük föl, majd futtassuk. Végtelen sokszor újra fog indulni, sokszor ki arra a konklúzióra fog jutni, hogy nem lehet root-olni, ilyenkor újra és újra indítsuk el. Ha a telefon újra indult, akkor ott fut a háttérben, csak hozzuk el?térbe. Addig indítgassuk újra, amíg nem root-olja a telefont.

Ha ez megvan indítsuk el a FolderMount alkalmazást, válasszuk ki az owncloud mappát a: /storage/emulated/0/owncloud, célmappa: /storage/external_SD/..., de ezt magától létrehozza.



A FolderMount magától el fog indulni minden telefon újraindításkor, és fel fogja mount-olni a beállított mappákat.

IPhone

Trouble shooting

Nem lehet megnyitni az Admin fület

Az Admin felhasználó nem tudja megnyitni a /settings/admin oldalt, a proci elkezd tekerni, és amíg nem csukjuk be a böngész?t nem tér magához a gép. Egy egy bug miatt van ha túl magas a logszint.

/var/www/html/owncloud/config/config.php

```
'logfile' => '/var/log/owncloud.log',
'loglevel' => 4, --->> ez itt nem lehet négy
'appstore.experimental.enabled' => false,
...
```