owncloud как subdomen

ownCloud – это свободное и открытое веб-приложение для синхронизации данных, общего доступа к файлам и удалённого хранения документов в «облаке».

Доступны клиенты для синхронизации данных с ПК под управлением Windows, OS X или Linux и с мобильными устройствами на iOS и Android. Кроме того, сохранённые данные доступны через веб-интерфейс ownCloud в любом браузере.

Доступ к облаку будет как субдомену, cloud.domen.com поэтому не забыть прописать в DNS.

Ставим последнюю стабильную версию:

```
# rpm --import https://download.owncloud.org/download/repositories/stable/CentOS_7/repodata/repomd.xml.key
# curl -L https://download.owncloud.org/download/repositories/stable/CentOS_7/ce:stable.repo -o /etc/yum.repos.d/ownCloud.repo
# yum install owncloud
```

Доустановим модули (если их нет) для owncloud:

```
$ sudo yum install php56w-xml php56w-gd php56w-intl php56w-mbstring
```

Сменим права и владельца:

```
# chmod -R 755 /var/www/html/owncloud
# chown -R nginx:www-data /var/www/html/owncloud
# chown -R nginx:www-data /var/lib/php/session
```

Установим memcached:

```
# yum install memcached php56w-pecl-memcache php56w-pecl-memcached php56w-pecl-apcu
```

```
# nano /etc/sysconfig/memcached
 PORT="11211"
```

```
 USER="memcached"
 MAXCONN="1024"
 CACHESIZE="64"
 OPTIONS="-l 127.0.0.1 -U 0"
```

Чтобы убедиться, что Memcached запущен и работает, введите следующее:

```
# memstat --servers="127.0.0.1"
 Server: 127.0.0.1 (11211)
 pid: 3831
 uptime: 9
 time: 1520028517
 version: 1.4.25
 ....
```

Устанавливаем Redis:

```
# yum install redis php56w-pecl-redis
```

Создаем конфиг для cloud пула:

```
# nano /etc/php-fpm.d/cloud.domen.com.conf

[cloud.tst-amo.net.ua]
listen = /var/run/php-fpm/cloud.domen.com.sock
listen.mode = 0666
user = nginx
group = www-data
chdir = /var/www/html/owncloud

# В зависимости от нагрузки меняем параметры
pm = dynamic
pm.max_children = 10
pm.start_servers = 2
pm.min_spare_servers = 2
pm.max_spare_servers = 4

# Default Value: clean env
env[HOSTNAME] = $HOSTNAME
env[PATH] = /usr/local/bin:/usr/bin:/bin
env[TMP] = /tmp
env[TMPDIR] = /tmp
```

```
env[TEMP] = /tmp
```

Создаем vhost cloud.domen.com:

```
# nano /etc/nginx/sites-available/owncloud

upstream php-handler {
    #server 127.0.0.1:9000;
    server unix:/var/run/php-fpm/cloud.domen.com.sock;
}

server {
    listen 80;
    server_name cloud.domen.com;

    # For Lets Encrypt, this needs to be served via HTTP
    location /.well-known/acme-challenge/ {
        root /var/www/html/owncloud; # Specify here where the
challenge file is placed
    }
    # enforce https
    location / {
        return 301 https://$server_name$request_uri;
    }
 }

server {
    listen 443 ssl http2;
    server_name cloud.domen.com www.cloud.domen.com;

    add_header X-Content-Type-Options nosniff;
    add_header X-Frame-Options "SAMEORIGIN";
    add_header X-XSS-Protection "1; mode=block";
    add_header X-Robots-Tag none;
    add_header X-Download-Options noopen;
    add_header X-Permitted-Cross-Domain-Policies none;

    include /etc/nginx/conf.d/ssl.conf;

    # Path to the root of your installation
    root /var/www/html/owncloud/;
    index index.php;
```

```
    location = /robots.txt {
         allow all;
         log_not_found off;
         access_log off;
    }

     # The following 2 rules are only needed for the
user_webfinger app.
    # Uncomment it if you're planning to use this app.
    #rewrite ^/.well-known/host-meta /public.php?service=host-
meta last;
              #rewrite   ^/.well-known/host-meta.json
/public.php?service=host-meta-json last;

    location = /.well-known/carddav {
          return 301 $scheme://$host/remote.php/dav;
    }
    location = /.well-known/caldav {
          return 301 $scheme://$host/remote.php/dav;
    }

    # set max upload size
    client_max_body_size 10G;
    fastcgi_buffers 64 4K; # Please see note 1
     fastcgi_ignore_headers X-Accel-Buffering; # Please see
note 2
    fastcgi_busy_buffers_size 192K;

    gzip off;

    # Uncomment if your server is build with the ngx_pagespeed
module
    # This module is currently not supported.
    #pagespeed off;

    error_page 403 /core/templates/403.php;
    error_page 404 /core/templates/404.php;

    location / {
       rewrite ^ /index.php$uri;
    }
```

```
                                        location         ~
^/(?:build|tests|config|lib|3rdparty|templates|data)/ {
       return 404;
    }
    location ~ ^/(?:\.|autotest|occ|issue|indie|db_|console) {
       return 404;
    }

                                        location         ~
^/(?:index|remote|public|cron|core/ajax/update|status|ocs/v[12
]|updater/.+|ocs-
provider/.+|core/templates/40[34])\.php(?:$|/) {
       fastcgi_split_path_info ^(.+\.php)(/.*)$;
       include fastcgi_params;
                        fastcgi_param   SCRIPT_FILENAME
$document_root$fastcgi_script_name;
        fastcgi_param SCRIPT_NAME $fastcgi_script_name; #
necessary for owncloud to detect the contextroot
https://github.com/owncloud/core/blob/v10.0.0/lib/private/AppF
ramework/Http/Request.php#L603
       fastcgi_param PATH_INFO $fastcgi_path_info;
       fastcgi_param HTTPS on;
       fastcgi_param modHeadersAvailable true; #Avoid sending
the security headers twice
       fastcgi_param front_controller_active true;
        fastcgi_read_timeout 180; # increase default timeout
e.g. for long running carddav/ caldav syncs with 1000+ entries
       fastcgi_pass php-handler;
       fastcgi_intercept_errors on;
        fastcgi_request_buffering off; #Available since NGINX
1.7.11
    }

    location ~ ^/(?:updater|ocs-provider)(?:$|/) {
        try_files $uri $uri/ =404;
        index index.php;
    }

    # Make sure it is BELOW the PHP block
    location ~ \.(?:css|js)$ {
        try_files $uri /index.php$uri$is_args$args;
```

```
        add_header Cache-Control "max-age=15778463";
        # Add headers to serve security related headers (It is
intended to have those duplicated to the ones above)
         # Before enabling Strict-Transport-Security headers
please read into this topic first.
           #add_header Strict-Transport-Security "max-
age=15552000; includeSubDomains";
        add_header X-Content-Type-Options nosniff;
        add_header X-Frame-Options "SAMEORIGIN";
        add_header X-XSS-Protection "1; mode=block";
        add_header X-Robots-Tag none;
        add_header X-Download-Options noopen;
        add_header X-Permitted-Cross-Domain-Policies none;
        # Optional: Don't log access to assets
        access_log off;
    }

                                        location         ~
\.(?:svg|gif|png|html|ttf|woff|ico|jpg|jpeg|map)$ {
       add_header Cache-Control "public, max-age=7200";
       try_files $uri /index.php$uri$is_args$args;
       # Optional: Don't log access to other assets
       access_log off;
    }
 }
```

```
# ln -s /etc/nginx/sites-available/owncloud /etc/nginx/sites-
enabled/
```

Проверяем и перезапускаем сервисы:

```
# nginx -t
# nginx -s reload
# service php-fpm restart
# service memcached restart
```

Заходим для дальнейшей настройки:

https://cloud.domen.com

После настройки приводим файл к такому виду:

```
# nano /var/www/html/owncloud/config/config.php
```

```
<?php
$CONFIG = array (
   'updatechecker' => false,
   'instanceid' => 'ocz4p432td1qkl',
   'passwordsalt' => 'IM1w1S8PbkjlwsbB4NcyW7cfT/pemjr',
   'secret' => '6Jxca0r8+zlklkml8Zi00TKfGY1gqeLxK9VeNOx',
   'trusted_domains' =>
   array (
     0 => 'cloud.domen.com',
   ),
   'datadirectory' => '/var/www/html/owncloud/data',
   'overwrite.cli.url' => 'https://cloud.domen.com',
   'dbtype' => 'mysql',
   'version' => '10.0.9.5',
   'dbname' => 'owncloud',
   'dbhost' => 'localhost',
   'dbtableprefix' => 'oc_',
   'dbuser' => 'owncloud',
   'dbpassword' => '9rtgu76yhgjkjghk8u',
   'logtimezone' => 'Europe/Kiev',
   'installed' => true,
   'filelocking.enabled' => 'true',
   'memcache.local' => '\\OC\\Memcache\\APCu',
   'memcache.distributed' => '\\OC\\Memcache\\Redis',
   'memcache.locking' => '\\OC\\Memcache\\Redis',
   'memcached_servers' =>
   array (
     0 =>
     array (
       0 => 'localhost',
       1 => 11211,
     ),
   ),
   'redis' =>
   array (
    'host' => 'localhost',
    'port' => 6379,
    'timeout' => 0,
    'password' => '',
    'dbindex' => 0,
   ),
```

```
   'mail_domain' => 'domen.com',
   'mail_from_address' => 'user',
   'mail_smtpmode' => 'smtp',
   'mail_smtpauthtype' => 'LOGIN',
   'mail_smtpauth' => 1,
   'mail_smtphost' => 'mail.domen.com',
   'mail_smtpport' => '25',
   'mail_smtpsecure' => 'tls',
   'mail_smtpname' => 'user',
   'mail_smtppassword' => 'user_password',
);
```

Перенесем для удобства обслуживания /var/www/html/owncloud/data в /home/www/owncloud/data

Вариант 1.

```
# mv /var/www/html/owncloud/data /var/www/html/owncloud/data_old
# cp -rp /var/www/html/owncloud/data_old /home/www/owncloud/data
# ln -s /home/www/owncloud/data /var/www/html/owncloud/
```

Или так:

Вариант 2.

Перемещаем хранилище файлов на выделенную партицию для этих целей:

```
$ sudo mkdir /ftp/owncloud
$ sudo chown nginx:www-data owncloud
$ sudo service nginx stop
$ sudo rsync -avz /var/www/html/owncloud/data /ftp/owncloud
$ sudo nano /var/www/html/owncloud/config/config.php

/* 'datadirectory' => '/var/www/html/owncloud/data', /*
 'datadirectory' => '/ftp/owncloud/data',
$ sudo service nginx start
```

Лог находится:
/var/www/html/owncloud/data/owncloud.log