owncloud

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

Предполагается, что установлена связка nginx, mariadb и php-fpm.

```
# yum install owncloud
# yum install memcached php56w-pecl-apcu redis php56w-pecl-redis
```

```
# systemctl start memcached
# systemctl enable memcached
# systemctl start redis
# systemctl enable redis
```

Проверяем:

```
[root@ring var]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address  Foreign Address State
PID/Program name
tcp        0      0 0.0.0.0:11211  0.0.0.0:*       LISTEN
19000/memcached
tcp        0      0 127.0.0.1:6379 0.0.0.0:*       LISTEN
18714/redis-server
udp             0          0  0.0.0.0:11211   0.0.0.0:*
19000/memcached
```

Чтобы убедиться, что экземпляр Memcached прослушивает локальный интерфейс 127.0.0.1, нужно проверить настройку по умолчанию в конфигурационном файле /etc/sysconfig/memcached, а также отключить прослушиватель UDP. Эти действия защитят сервер от атак типа «отказ в обслуживании».

После установки правим:

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

```
USER="memcached"
MAXCONN="1024"
CACHESIZE="256"
OPTIONS="-U 0 -l 127.0.0.1"
```

Настройки *vhost* nginx:

```
$ cat /etc/nginx/sites-available/cloud.example.com

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

server {
 listen 80;
 server_name cloud.example.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.example.com www.cloud.example.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;
 }
 }

$ cat /etc/php-fpm.d/cloud.example.com.conf
[cloud.example.com]
listen = /var/run/php-fpm/php-fpm.cloud.example.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
```

Конфигурационный файл owncloud:

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

```
<?php
$CONFIG = array (
       'updatechecker' => false,
       'instanceid' => 'bjkkjljkf48o5',
       'passwordsalt' => '3BMoYgjhghhkStcX6Iw4lgT7lgh',
                                              'secret'       =>
'U1ZEBI3W9PppZNFPIDPkJJsizVAi3eox9ZvdYqT6JQkHRLZP',
       'trusted_domains' =>
 array (
       0 => 'host.example.com',
 ),
 'datadirectory' => '/var/www/html/owncloud/data',
 'overwrite.cli.url' => 'https://host.example.com/owncloud',
 'dbtype' => 'mysql',
 'version' => '10.0.8.5',
 'dbname' => 'owncloud',
 'dbhost' => 'localhost',
 'dbtableprefix' => 'oc_',
 'dbuser' => 'owncloud',
 'dbpassword' => 'xxxxxxxxxxx',
 '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(
        array('localhost', 11211),
 ),
 'redis' => [
         'host' => 'localhost', // Can also be a unix domain
socket => '/tmp/redis.sock'
         'port' => 6379,
         'timeout' => 0,
          'password' => '',  // Optional, if not defined no
password will be used.
         'dbindex' => 0     // Optional, if undefined SELECT
will not run and will
                            // use Redis Server's default DB
Index.
 ],
);
```

Чтобы убрать ошибки на вкладке *Настройки –> Основные*

Переключаем Cron (Планировщик) в Cron (системный) и в терминале набираем:

```
# crontab -e
*/15 * * * * nginx php /var/www/html/owncloud/cron.php
```

[Performance tips for Redis Cache Server]

3. Disable THP

1417:M 25 Oct 06:13:31.840 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command ‘echo never > /sys/kernel/mm/transparent_hugepage/enabled’ as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled
This is also simple to fix by just running the recommended command as stated in the warning.

```
echo never > /sys/kernel/mm/transparent_hugepage/enabled
```

Go to /etc/rc.local

$~: sudo nano /etc/rc.local

Add this:

echo never > /sys/kernel/mm/transparent_hugepage/enabled

Now this will be persistent upon reboot as well.