rtorrent+rutorrent en contenedor

instalacion rtorrent+rutorrent

1. crear estructura de directorios:

```
mkdir -p data/downloads/completo
mkdir -p data/root
mkdir -p data/rutorrent
```

2. copiar ficheros de configuración contenedor en **/root** (sacados del propio contenedor y modificar los ficheros de configuración de nginx):
   - añadir:

```
location /completo {
        alias /downloads/completo;
        autoindex on;
}
```

3. copiar fichero **config.php** en **data/rutorrent** (para cambiar variable `$forbidUserSettings = true;`)
4. crear enlaces simbólicos:

```
# los enlaces funcionarán desde dentro del contenedor por el mapeo de volumen
ln -s /etc/letsencrypt/live/k0.vividumcodex.com/cert.pem nginx.crt
ln -s /etc/letsencrypt/live/k0.vividumcodex.com/privkey.pem nginx.pem
```

5. ejecutar:

```
CERTBOT_PATH="/home/user/workspaces/nginx+certbot"

docker run \
    -dt \
    --restart unless-stopped \
    --name rtorrent-rutorrent \
    -p 443:443 \
    -p 49160:49160/udp \
    -p 49161:49161 \
    -v ${PWD}/data/downloads:/downloads \
    -v ${PWD}/data/root:/root \
    -v ${PWD}/data/rutorrent/config.php:/var/www/rutorrent/conf/config.php \
    -v ${CERTBOT_PATH}/data/certbot/conf:/etc/letsencrypt \
    -v ${CERTBOT_PATH}/data/certbot/www:/var/www/certbot \
diameter/rtorrent-rutorrent:latest
```

pasos básicos

- montar estructura de carpetas
  - downloads : ubicación descargas
  - root : ubicación archivos configuración NGINX/RTORRENT/RUTORRENT modificados
- ejecutar docker según script

run.sh

```
STORAGE_PATH=/home/user/storage/rtorrent+rutorrent

docker run \
    -dt \
    --name rtorrent-rutorrent \
    --restart unless-stopped \
    -p 80:80 \
    -p 443:443 \
    -p 49160:49160/udp \
    -p 49161:49161 \
    -v ${STORAGE_PATH}/downloads:/downloads \
    -v ${STORAGE_PATH}/root:/root \
diameter/rtorrent-rutorrent:latest
```

crypt.pl

```
#!/usr/bin/perl
use strict;

chomp(my $filename=$ARGV[0]);
chomp(my $username=$ARGV[1]);
chomp(my $password=$ARGV[2]);

if (!$filename || !$username || !$password) {
  print "USAGE: ./crypt.pl filename username password\n\n";
} else {
  open my $fh, ">>", $filename or die $!;
  print $fh $username . ":" . crypt($password, $username) . "\n";
  close $fh or die $!;
}
```

```
# You may add here your
# server {
#   ...
# }
# statements for each of your virtual hosts to this file

##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# http://wiki.nginx.org/Pitfalls
# http://wiki.nginx.org/QuickStart
# http://wiki.nginx.org/Configuration
#
# Generally, you will want to move this file somewhere, and start with a clean
# file but keep this around for reference. Or just disable in sites-enabled.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##
```

```
server {
    listen 443 ssl;
  listen [::]:443 default_server ipv6only=on;

        keepalive_timeout   60;
        ssl_certificate      /etc/nginx/ssl/nginx.crt;
        ssl_certificate_key  /etc/nginx/ssl/nginx.key;
                ssl_ciphers "AES128+EECDH:AES128+EDH";
                ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
                ssl_prefer_server_ciphers on;
                ssl_session_cache shared:SSL:10m;
                add_header X-Frame-Options SAMEORIGIN;
                add_header X-Content-Type-Options nosniff;

    root /var/www/rutorrent;
    index index.php index.html index.htm;

    # Make site accessible from http://localhost/
    server_name localhost;

#   location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
        # Uncomment to enable naxsi on this location
        # include /etc/nginx/naxsi.rules
                auth_basic "Privat";
                auth_basic_user_file /var/www/rutorrent/.htpasswd;
#   }

    # Only for nginx-naxsi used with nginx-naxsi-ui : process denied requests
    #location /RequestDenied {
    #    proxy_pass http://127.0.0.1:8080;
    #}

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    #error_page 500 502 503 504 /50x.html;
    #location = /50x.html {
    #    root /usr/share/nginx/html;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #
    location ~ \.php$ {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
    # NOTE: You should have "cgi.fix_pathinfo = 0;" in php.ini

        # With php5-cgi alone:
    #    fastcgi_pass 127.0.0.1:9000;
    #    # With php5-fpm:
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
        fastcgi_index index.php;
        include fastcgi_params;
```

```
                fastcgi_param HTTPS on;
                fastcgi_param SCRIPT_FILENAME $request_filename;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}

    location /RPC2 {
            include scgi_params;
            scgi_pass 127.0.0.1:5000;
                scgi_param SCRIPT_NAME /RPC2;
    }

    location /completo {
#        try_files $uri $uri/ =404;
#       Auth_basic "Restricted Complete";
#        auth_basic_user_file /var/www/rutorrent/.htpasswd;
        alias /downloads/complete;
        autoindex on;
    }

        location /well-known {
                auth_basic "off";
                alias /well-known;
        }
}
```

help.md

```
# rtorrent + rutorrent
[dockerhub] (https://hub.docker.com/r/diameter/rtorrent-rutorrent/)

## notas de run
- Web UI ports: 80 and 443 (can be remapped in 'docker run')
- DHT UDP port: 49160 (can be remapped)
- Incoming connections port: 49161 (can be remapped)
- Downloads volume: /downloads
- rtorrent scratch files (.rtorrent/{watch|session} will be created
automatically): /downloads
- autodl-irssi config files are created automatically: /downloads/.autodl
- external rtorrent config (.rtorrent/.rtorrent.rc): /downloads
- external ruTorrent ui config (config will be created automatically):
/downloads/.rutorrent
- external nginx and rtorrent logs: /downloads/.log/
- rtorrent uid and gid: USR_ID and GRP_ID env vars, default is 1000:1000
- php-fpm memory limit: PHP_MEM env var, default is 256M
- disable IPv6 binding in nginx: set env var NOIPV6=1, default is not set
- alternative webroot: WEBROOT env var, defailt is /

## htpasswd
```

```
Put .htpasswd into your /downloads volume root, the container will re-read
.htpasswd each time it starts. To remote auth, simply remove .htpasswd and
restart your container.
### generate
- `printf "John:$(openssl passwd -crypt V3Ry)\n" >> .htpasswd # this example
uses crypt encryption`
- `printf "Mary:$(openssl passwd -apr1 SEcRe7)\n" >> .htpasswd # this example
uses apr1 (Apache MD5) encryption`
- `printf "Jane:$(openssl passwd -1 V3RySEcRe7)\n" >> .htpasswd # this example
uses MD5 encryption`
- ```(PASSWORD="SEcRe7PwD";SALT="$(openssl rand -base64 3)";SHA1=$(printf
"$PASSWORD$SALT" | openssl dgst -binary -sha1 | xxd -ps | sed 's#$#'"`echo -n
$SALT | xxd -ps`"'#' | xxd -r -ps | base64);printf "Jim:{SSHA}$SHA1\n" >>
.htpasswd) # this example uses SSHA encryption```
- script *crypt.pl*

## TLS
Put your keyfile (shall be named nginx.key) and your certificate (nginx.crt)
into /dowloads volume root, the container looks for these files each time it
starts.
```

+ info

- https://www.rapidseedbox.com/kb/beginners-guide-to-rutorrent
- https://github.com/Novik/ruTorrent/wiki/Config

From:
https://miguelangel.torresegea.es/wiki/ - **miguel angel torres egea**

Permanent link:
https://miguelangel.torresegea.es/wiki/linux:debian:seedbox:rtorrent_rutorrent?rev=1580595457

Last update: **01/02/2020 14:17**