

Nextcloud

- Nextcloud & Nginx

Nextcloud Nginx

```
map $arg_v $asset_immutable {
    "" "";
    default "", immutable";
}

server {
    listen 80;
    listen [::]:80;
    server_name cloud.site.com;

    # Prevent nginx HTTP Server Detection
    server_tokens off;

    # Enforce HTTPS
    return 301 https://$server_name$request_uri;
}

server {
    listen 443    ssl http2;
    listen [::]:443 ssl http2;
    server_name cloud.zfun.com.tw;

    # Path to the root of your installation
    root /var/www/cloud;

    # Use Mozilla's guidelines for SSL/TLS settings
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/
    ssl_certificate    /etc/ssl/nginx/cloud.example.com.crt;
    ssl_certificate_key /etc/ssl/nginx/cloud.example.com.key;

    # Prevent nginx HTTP Server Detection
    server_tokens off;

    # HSTS settings
    # WARNING: Only add the preload option once you read about
```

```
# the consequences in https://hstspreload.org/. This option
# will add the domain to a hardcoded list that is shipped
# in all major browsers and getting removed from this list
# could take several months.
#add_header Strict-Transport-Security "max-age=15768000; includeSubDomains; preload" always;

# set max upload size and increase upload timeout:
client_max_body_size 512M;
client_body_timeout 300s;
fastcgi_buffers 64 4K;

# Enable gzip but do not remove ETag headers
gzip on;
gzip_vary on;
gzip_comp_level 4;
gzip_min_length 256;
gzip_proxied expired no-cache no-store private no_last_modified no_etag auth;
gzip_types application/atom+xml text/javascript application/javascript application/json application/ld+json
application/manifest+json application/rss+xml application/vnd.geo+json application/vnd.ms-fontobject
application/wasm application/x-font-ttf application/x-web-app-manifest+json application/xhtml+xml
application/xml font/opentype image/bmp image/svg+xml image/x-icon text/cache-manifest text/css text/plain
text/vcard text/vnd.rim.location.xloc text/vtt text/x-component text/x-cross-domain-policy;

# Pagespeed is not supported by Nextcloud, so if your server is built
# with the `ngx_pagespeed` module, uncomment this line to disable it.
#pagespeed off;

# The settings allows you to optimize the HTTP2 bandwidth.
# See https://blog.cloudflare.com/delivering-http-2-upload-speed-improvements/
# for tuning hints
client_body_buffer_size 512k;

# HTTP response headers borrowed from Nextcloud `.htaccess`
add_header Referrer-Policy          "no-referrer"      always;
add_header X-Content-Type-Options    "nosniff"         always;
add_header X-Frame-Options           "SAMEORIGIN"       always;
add_header X-Permitted-Cross-Domain-Policies "none"      always;
add_header X-Robots-Tag               "noindex, nofollow" always;
add_header X-XSS-Protection          "1; mode=block"   always;

# Remove X-Powered-By, which is an information leak
```

```
fastcgi_hide_header X-Powered-By;
```

```
# Set .mjs and .wasm MIME types
```

```
# Either include it in the default mime.types list
```

```
# and include that list explicitly or add the file extension
```

```
# only for Nextcloud like below:
```

```
include mime.types;
```

```
types {
```

```
    # text/javascript js mjs;
```

```
    text/javascript mjs;
```

```
    application/wasm wasm;
```

```
}
```

```
# Specify how to handle directories -- specifying `/index.php$request_uri`
```

```
# here as the fallback means that Nginx always exhibits the desired behaviour
```

```
# when a client requests a path that corresponds to a directory that exists
```

```
# on the server. In particular, if that directory contains an index.php file,
```

```
# that file is correctly served; if it doesn't, then the request is passed to
```

```
# the front-end controller. This consistent behaviour means that we don't need
```

```
# to specify custom rules for certain paths (e.g. images and other assets,
```

```
# `/updater`, `/ocs-provider`), and thus
```

```
# `try_files $uri $uri/ /index.php$request_uri`
```

```
# always provides the desired behaviour.
```

```
index index.php index.html /index.php$request_uri;
```

```
# Rule borrowed from `.htaccess` to handle Microsoft DAV clients
```

```
location = / {
```

```
    if ( $http_user_agent ~ ^DavClnt ) {
```

```
        return 302 /remote.php/webdav/$is_args$args;
```

```
    }
```

```
}
```

```
location = /robots.txt {
```

```
    allow all;
```

```
    log_not_found off;
```

```
    access_log off;
```

```
}
```

```
# Make a regex exception for `/well-known` so that clients can still
```

```
# access it despite the existence of the regex rule
```

```
# `location ~ /\.([autotest|...])` which would otherwise handle requests
```

```

# for `/.well-known`.
location ^~ /.well-known {
    # The rules in this block are an adaptation of the rules
    # in `.htaccess` that concern `/.well-known`.

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

    location /.well-known/acme-challenge { try_files $uri $uri/ =404; }
    location /.well-known/pki-validation { try_files $uri $uri/ =404; }

    # Let Nextcloud's API for `/.well-known` URIs handle all other
    # requests by passing them to the front-end controller.
    return 301 /index.php$request_uri;
}

# Rules borrowed from `.htaccess` to hide certain paths from clients
location ~ ^/(?!build|tests|config|lib|3rdparty|templates|data)(?:$|/) { return 404; }
location ~ ^/(?!\.|autotest|occ|issue|indie|db_|console) { return 404; }

# Ensure this block, which passes PHP files to the PHP process, is above the blocks
# which handle static assets (as seen below). If this block is not declared first,
# then Nginx will encounter an infinite rewriting loop when it prepends `/index.php`
# to the URI, resulting in a HTTP 500 error response.
location ~ \.php(?:$|/) {
    # Required for legacy support
    rewrite ^/(?!index|remote|public|cron|core|ajax|update|status|ocs/v[12]|updater/.+|ocs-provider/.+|richdocumentscode(_arm64)?\proxy) /index.php$request_uri;

    fastcgi_split_path_info ^(.+?\.php)(/.*)$;
    set $path_info $fastcgi_path_info;

    try_files $fastcgi_script_name =404;

    include fastcgi_params;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $path_info;
    fastcgi_param HTTPS on;

    fastcgi_param modHeadersAvailable true;      # Avoid sending the security headers twice
    fastcgi_param front_controller_active true;  # Enable pretty urls

```

```

# fastcgi_pass php-handler;
fastcgi_pass unix:/var/run/php/php8.2-fpm.sock;


fastcgi_intercept_errors on;
fastcgi_request_buffering off;


fastcgi_max_temp_file_size 0;
}


# Serve static files
location ~ \.(?:css|js|mjs|svg|gif|png|jpg|ico|wasm|tflite|map|ogg|flac)$ {
    try_files $uri /index.php$request_uri;
    # HTTP response headers borrowed from Nextcloud `.htaccess`
    add_header Cache-Control          "public, max-age=15778463$asset_immutable";
    add_header Referrer-Policy        "no-referrer"      always;
    add_header X-Content-Type-Options "nosniff"          always;
    add_header X-Frame-Options        "SAMEORIGIN"       always;
    add_header X-Permitted-Cross-Domain-Policies "none"    always;
    add_header X-Robots-Tag            "noindex, nofollow" always;
    add_header X-XSS-Protection       "1; mode=block"    always;
    access_log off;    # Optional: Don't log access to assets
}


location ~ /\.woff2?$ {
    try_files $uri /index.php$request_uri;
    expires 7d;    # Cache-Control policy borrowed from `.htaccess`
    access_log off;    # Optional: Don't log access to assets
}


# Rule borrowed from `.htaccess`
location /remote {
    return 301 /remote.php$request_uri;
}


location / {
    try_files $uri $uri/ /index.php$request_uri;
}
}

```