Nextcloud

网盘

安装环境依赖

```
sudo apt update && sudo apt upgrade
sudo apt install mariadb-server php-gd php-mysql php-curl php-mbstring php-intl php-gmp php-xml php-imagick php-zip
```

SQL配置

```
#登入msql
mysql -u root -p
#创建nextcloud_db数据库
CREATE DATABASE IF NOT EXISTS nextcloud_db CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
CREATE USER 'admin'@'localhost' IDENTIFIED BY '********';
GRANT ALL PRIVILEGES ON nextcloud_db.* TO 'admin'@'localhost';
FLUSH PRIVILEGES;
quit;
```

Nginx配置

NGINX configuration — Nextcloud latest Administration Manual latest documentation

https://docs.nextcloud.com/server/latest/admin_manual/installation/nginx.html

nextcloud.conf

```
# Version 2025-07-23

upstream php-handler {
    #server 127.0.0.1:9000;
    server unix:/run/php/php8.4-fpm.sock;
}

# Set the `immutable` cache control options only for assets with a cache busting `v` argument
map $arg_v $asset_immutable {
    "" "";
    default ", immutable";
}
```

```
server {
    listen 80;
    listen [::]:80;
    server_name cloud.example.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;
    # With NGinx >= 1.25.1 you should use this instead:
    listen 443      ssl;
    listen [::]:443 ssl;
    http2 on;
    server_name cloud.example.com;

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

    # Use Mozilla's guidelines for SSL/TLS settings
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/
    ssl_certificate     /etc/nginx/conf.d/cert/cloud.example.com.pem;
    ssl_certificate_key /etc/nginx/conf.d/cert/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=31536000;
includeSubDomains; preload" always;

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

    # Proxy and client response timeouts
    # Uncomment an increase these if facing timeout errors during large
file uploads
    #proxy_connect_timeout 60s;
    #proxy_send_timeout 60s;
```

```
    #proxy_read_timeout 60s;
    #send_timeout 60s;

    # 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;

    # 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 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(?:$|/) {
        # Required for legacy support
        rewrite
^/(?!index|remote|public|cron|core\/ajax\/update|status|ocs\/v[12]|upda
ter\/.+|ocs-provider\/.+|.+\/richdocumentscode(_arm64)?\/proxy)
/index.php$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_intercept_errors on;
        fastcgi_request_buffering on;                   # Required as
PHP-FPM does not support chunked transfer encoding and requires a valid
ContentLength header.

        # PHP-FPM 504 response timeouts
        # Uncomment and increase these if facing timeout errors during
large file uploads
        #fastcgi_read_timeout 60s;
        #fastcgi_send_timeout 60s;
        #fastcgi_connect_timeout 60s;

        fastcgi_max_temp_file_size 0;
    }

    # Serve static files
    location ~
\.(?:css|js|mjs|svg|gif|ico|jpg|png|webp|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;
        access_log off;     # Optional: Don't log access to assets
    }

    location ~ \.(otf|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;
    }
```

```
}
```

Nextcloud部署

现在下载最新 Nextcloud 版本的存档：

转到 Nextcloud 安装页面。

转到下载 Server > Community Projects然后下载 tar.bz2 或 .zip 存档。

这将下载名为 nextcloud-x.y.z.tar.bz2 或 nextcloud-x.y.z.zip 的文件 （其中 x.y.z 是版本号）。

下载其相应的校验和文件，例如 nextcloud-x.y.z.tar.bz2.md5 或 nextcloud-x.y.z.tar.bz2.sha256

验证 MD5 或 SHA256 总和：

```
scp .\nextcloud.zip user@127.0.0.1:/home/user

unzip nextcloud.zip
sudo mv ./nextcloud /var/www/
sudo chown -R www-data:www-data /var/www/nextcloud
```

浏览器访问向导安装

管理员 admin

密码 123456

路径默认 /var/www/nextcloud/data

数据库用户 admin

数据库密码 123456

数据库名称 nextcloud_db

数据库地址 localhost:3306