# Nextcloud家用搭建

## 准备阶段

系统:rocky linux 9

环境: 已关闭swap，selinux还有防火墙。

需要使用的软件:

Podman-compose或者docker-compose

Server version: 10.11.12-MariaDB MariaDB Server

Redis server v=8.0.1。

Nginx 1.20.1

### MariaDB部份

允许root用户通过外部访问MariaDB:

找到MariaDB配置文件，修改 bind-address 为 0.0.0.0。

登录mysql运行以下命令

创建root用户密码

ALTER USER 'root'@'localhost' IDENTIFIED BY 'new\_password';

FLUSH PRIVILEGES;

允许外部访问

GRANT ALL PRIVILEGES ON \*.\* TO 'root '@'%' IDENTIFIED BY '密码 ';

FLUSH PRIVILEGES;

然后重启数据库。

使用navicat工具连接数据库后创建nextcloud需要使用的数据库和账号。

### Redis部份

允许redis通过外部访问，找到redis配置文件

修改bind 为 0.0.0.0。

添加配置

requirepass 123456 （这里是设置redis的访问密码）

maxmemory 1048576000

timeout 65

### podm-compose部份

创建一个nextcloud的yaml文件。

version: '3.8'  
services:  
 nextcloud:  
 image: nextcloud:latest  
 container\_name: nextcloud  
 restart: always  
 ports:  
 - 10000:80  
  
 volumes:  
 - /home/nextcloud:/var/www/html

## 步骤

### 第一步nextcloud镜像拉取

使用podman-compose拉取镜像，找到刚刚存放nextcloud.yaml的目录，

我一般是放在home目录下，cd /home以后输入

podman-compose -f nextcloud.yaml up -d

等待镜像下载并启动。

镜像启动完成后输入服务器ip加端口号10000进入nextcloud的初始化界面

输入创建好的数据库名字和账号密码，然后创建nextcloud的管理员账号。

### 第二步nextcloud优化

进入home目录找到config.php文件，打开编辑，添加redis配置

'memcache.distributed' => '\\OC\\Memcache\\Redis',  
'memcache.locking' => '\\OC\\Memcache\\Redis',  
'redis' =>   
 **array** (  
 'host' => '127.0.0.1', #这里填写你的redis ip地址  
 'password' => 'password', #这里填写你设置的redis密码  
 'port' => 6379,  
 'timeout' => 1.0,  
 'dbindex' => 0,  
 ),

其他优化

'default\_phone\_region' => 'CN',  
 'upload\_max\_filesize' => '20g',  
 'post\_max\_size' => '20g',

然后保存退出。

Nextcloud两个警告解决命令

出现one or mimetype 使用

podman exec -it -u root nextcloud php occ maintenance:repair --include-expensive

邮件应用等缺乏索引用下面的命令

podman exec -it -u root nextcloud php occ db:add-missing-indices

解除上传分块大小限制

podman exec -it -u root nextcloud php occ config:app:set files max\_chunk\_size --value 0

nextcloud cron定时任务

手动运行 podman exec -u root nextcloud php -f /var/www/html/cron.php

然后输入crontab -e添加定时任务

\*/5 \* \* \* \* podman exec -u root nextcloud php -f /var/www/html/cron.php

修改php配置

podman exec -it -u root nextcloud bash 进入容器

cd /usr/local/etc/php/conf.d

echo "upload\_max\_filesize = 20G" >> /usr/local/etc/php/conf.d/upload.ini

echo "post\_max\_size = 20G" >> /usr/local/etc/php/conf.d/upload.ini

echo "memory\_limit = 1G" >> /usr/local/etc/php/conf.d/upload.ini

echo "max\_execution\_time = 0" >> /usr/local/etc/php/conf.d/upload.ini

echo "max\_input\_time = 3600" >> /usr/local/etc/php/conf.d/upload.ini

修改OPcache配置

vi opcache-recommended.ini

提示没有vi可以apt装一个vim。 apt update apt install vim

opcache.enable=1

opcache.interned\_strings\_buffer=32

opcache.max\_accelerated\_files=10000

opcache.memory\_consumption=512

opcache.save\_comments=1

opcache.revalidate\_freq=0

opcache.jit=1255

opcache.jit\_buffer\_size=128

opcache.validate\_timestamps=0

nextcloud查看视频预览图

安装ffmpeg，还是使用命令进入容器

podman exec -it -u root nextcloud bash

然后apt install ffmpeg

在config.php里面添加

'enable\_previews' => true,  
 'enabledPreviewProviders' =>  
 **array** (  
 0 => 'OC\\Preview\\Image',  
 1 => 'OC\\Preview\\Movie',  
 2 => 'OC\\Preview\\TXT',  
 3 => 'OC\\Preview\\MKV',  
 4 => 'OC\\Preview\\MP4',  
 5 => 'OC\\Preview\\AVI',  
),

### 第三步Nginx配置和config.php对应配置

#### Nginx配置

server {

    listen 8889 ssl http2;

    server\_name cloud.nextcloud.com; #你的域名

    # SSL强化配置 这里你的ssl证书目录

    ssl\_certificate /home/nextcloud/ssl/fullchain.cer;

    ssl\_certificate\_key /home/nextcloud/ssl/cert.key;

    ssl\_protocols TLSv1.2 TLSv1.3;

    ssl\_ciphers 'ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384';

    ssl\_prefer\_server\_ciphers on;

    ssl\_session\_cache shared:SSL:10m;

    ssl\_session\_timeout 1h;

    ssl\_session\_tickets off;

    ssl\_buffer\_size 4k;

    # 安全头

    add\_header Strict-Transport-Security "max-age=15552000; includeSubDomains; preload";

    add\_header Permissions-Policy "geolocation=(), microphone=(), camera=()";

    add\_header Referrer-Policy "strict-origin-when-cross-origin";

    # 文件上传

    client\_max\_body\_size 20G;

    client\_body\_timeout 300s;

    client\_header\_timeout 60s;

    # 代理通用配置

    proxy\_http\_version 1.1;

    proxy\_set\_header Connection "";

    proxy\_set\_header X-Forwarded-Proto $scheme;

    proxy\_set\_header X-Forwarded-Ssl on;

    # WebDAV支持

    location ~ ^/.well-known/(carddav|caldav) {

        return 301 https://$host:$server\_port/remote.php/dav;

    }

    # 主代理配置

    location / {

        proxy\_pass http://127.0.0.1:10000;

        proxy\_set\_header Host $http\_host;

        proxy\_set\_header X-Real-IP $remote\_addr;

        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

        # 性能优化

        proxy\_request\_buffering off;

        proxy\_max\_temp\_file\_size 0;

        proxy\_connect\_timeout 3600;

        proxy\_send\_timeout 3600;

        proxy\_read\_timeout 3600;

        # 隐藏敏感头

        proxy\_hide\_header X-Powered-By;

        proxy\_hide\_header Server;

    }

    # 安全限制

    location ~ ^/(\.ht|data|config|db\_structure\.xml|README) {

        deny all;

    }

    # 压缩配置

    gzip on;

    gzip\_vary on;

    gzip\_min\_length 1k;

    gzip\_types text/plain text/css application/json application/javascript text/xml application/xml application/xml+rss text/javascript;

    gzip\_proxied any;

    gzip\_comp\_level 6;

    # 日志

    access\_log /var/log/nginx/nextcloud.access.log combined buffer=64k flush=5m;

    error\_log /var/log/nginx/nextcloud.error.log warn;

}

#### Config.php需要添加的配置

'trusted\_domains' =>

array (

0 => '192.168.2.6:10000', #这是一般是默认不用动的，我的服务器IP是192.168.26

1 => 'cloud.nextcloud.com', #这里添加你的域名

）

'trusted\_proxies' =>

array (

0 => '192.168.2.6',

1 => '127.0.0.1',

),

'forwarded\_for\_headers' =>

array (

0 => 'HTTP\_X\_FORWARDED\_FOR',

1 => 'HTTP\_X\_REAL\_IP',

),

'overwritewebroot' => '',

'overwriteprotocol' => 'https',

'maintenance\_window\_start' => 1,

'cookie\_secure' => true,

'session.cookie\_samesite' => 'None',

'session.cookie\_secure' => true,