Nextcloud

Install Nextcloud on Docker

Version 2 configuration works best on the latest Nextcloud v29.0.

As of this version, when it gives you the first admin login, create your new login and wait for a bit. You may need refresh or, leave and go back to the page.
It may take a while but it SHOULD eventually end up giving you the login screen to enter your new credentials.

MIGRATIONS: if you need to migrate, backup or restore, you can still use the version 2 config but read the migration wiki before you continue.

Version 1:

```
version: '3'

#volumes:
#  nextcloud:
#  mariadb:

services:
  mariadb:
    image: mariadb
    container_name: mariadb
    restart: always
    command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
    volumes:
      - /mounted/local/storage/location/data:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=Mariaisabitch
      - MYSQL_PASSWORD=Mariaisabitch
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=marialikescock

  nextcloud:
    image: nextcloud
    container_name: nextcloud
    restart: always
    links:
```

```
      - mariadb
    volumes:
      - /mounted/local/storage/location/config:/config
      - /mounted/local/storage/location/data:/data
      - /mounted/local/storage/location/apps:/apps
    ports:
      - 8091:80
    environment:
      - MYSQL_PASSWORD=Mariaisabitch
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=marialikescock
      - MYSQL_HOST=mariadb
```

Version 2:

```
version: '3'
services:
  mariadb:
   image: mariadb
   container_name: nextmaria
   restart: unless-stopped
   command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
   volumes:
     - /mounted/local/storage/location/datadb:/var/lib/mysql
   ports:
     - 3006:3306
   environment:
     - MYSQL_ROOT_PASSWORD=Mariaisfat
     - MYSQL_PASSWORD=Mariaisfat
     - MYSQL_DATABASE=nextcloud
     - MYSQL_USER=nextcloud

  Nextcloud:
    image: nextcloud
    container_name: nextcloud
    restart: unless-stopped
    links:
      - mariadb
    volumes:
```

```
    - /mounted/local/storage/location/data:/var/www/html
  ports:
    - 8000:80
  environment:
    - MYSQL_PASSWORD=Mariaisfath
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud
    - MYSQL_HOST=mariadb
```

Install Nextcloud with Apache 2 | locally

Manual install using Apache 2

Get zip file
`wget [https://download.nextcloud.com/server/releases/nextcloud-22.2.0.zip](https://download.nextcloud.com/server/releases/nextcloud-22.2.0.zip)` (find the lates version)

Unzip it: Download unzip if needed
`unzip name.zip` then file name.zip

Change ownership of Nextcloud directory so apache is the owner
`sudo chown www-data:www-data -R nextcloud`

Move nextcloud dir to the location Apache searches for its own folders
`sudo mv nextcloud /var/www/html/`

Create configuration file for that directory (using APACHE2)
`sudo nano /etc/apache2/sites-available/nextcloud.conf`

Use the following and edit whats needed

```
<VirtualHost *:80>
        DocumentRoot ****/var/www/html/nextcloud/
        ServerName familystronhold.onthewifi.com

 <Directory /var/www/html/nextcloud/>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
            <IfModule mod_dav.c>
               Dav off
            </IfModule>
            SetEnv HOME /var/www/html/nextcloud
            SetEnv HTTP_HOME /var/www/html/nextcloud
    </Directory>
```

```
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined


</VirtualHost>
```

Enable nextcould config file for Apache

```
sudo a2ensite nextcloud.conf
```

Restart apache2

```
sudo systemctl restart apache2
```

Disable test page

```
sudo a2dissite 000-default
```

Enable some modules on apache2 for nextcloud

```
sudo a2enmod rewrite headers env dir mime
```

Restart apache2 again

Install a certificate bot REPO for "lets encrypt"

Go to: https://certbot.eff.org/lets-encrypt/ubuntufocal-apache

Migrate Nextcloud To Another Server

Restoring from backup. Use this same method for migrating.

1) You use Rsync to move the backed up files to the new server. It will keep the tags and other permission that the backup was oriniglally stored with. If you don't care about that, just copy it. (rsync s pretty quick and copying either way).

2) I found it is easier to start a new instance of your new next cloud, then, copy over your data. This is because some files that are too old, dont sync well with the latest versions and, we are also getting rid of anything else that is not compatible... for whatever reasons. So... lets get started.

First things first. Where are you storing you volumes? I am putting them here:

`/home/server/nextcloud/`

I base everything off of the yml files. You can use hub.docker.com or you can create your own. I did a little of both here.

Create 2 new folders/directories in that same location for the 2 new volumes...
if i am in the `/home/server/nextcloud/` directory I would type:

```
sudo mkdir data datadb
```

Put your yml file in here `/home/server/nextcloud` or anywhere you want to keep that project file at.

I am going to copy over the database to the new server

```
rsync -Aax /backup/directory/nextcloud/datadb /home/server/nextcloud/datadb
```

I am going to NOT move the data files yet. We have a /data folder ready to go and empty for nextcloud to fill up.

Start up your new nextcloud now.
-Make sure it is running
-Make sure the site is up and it gives you the sign in screen.

Create a new login

All of your old info will be stored/saved however this requires you to create a new admin account. Later, you can remove it using one of your original user names.

While creating this, sometimes its slow. Sometimes its not.

> You may want to speed this up by going over to your php stuff.

It didnt work if:

If, during sign in, if you see it's asking for database info, it didnt work.

You dont need to wait for it to load. If it looks like it is starting to install stuff, you can leave. We just wanted it to have enough info to start creating the new config.php file.

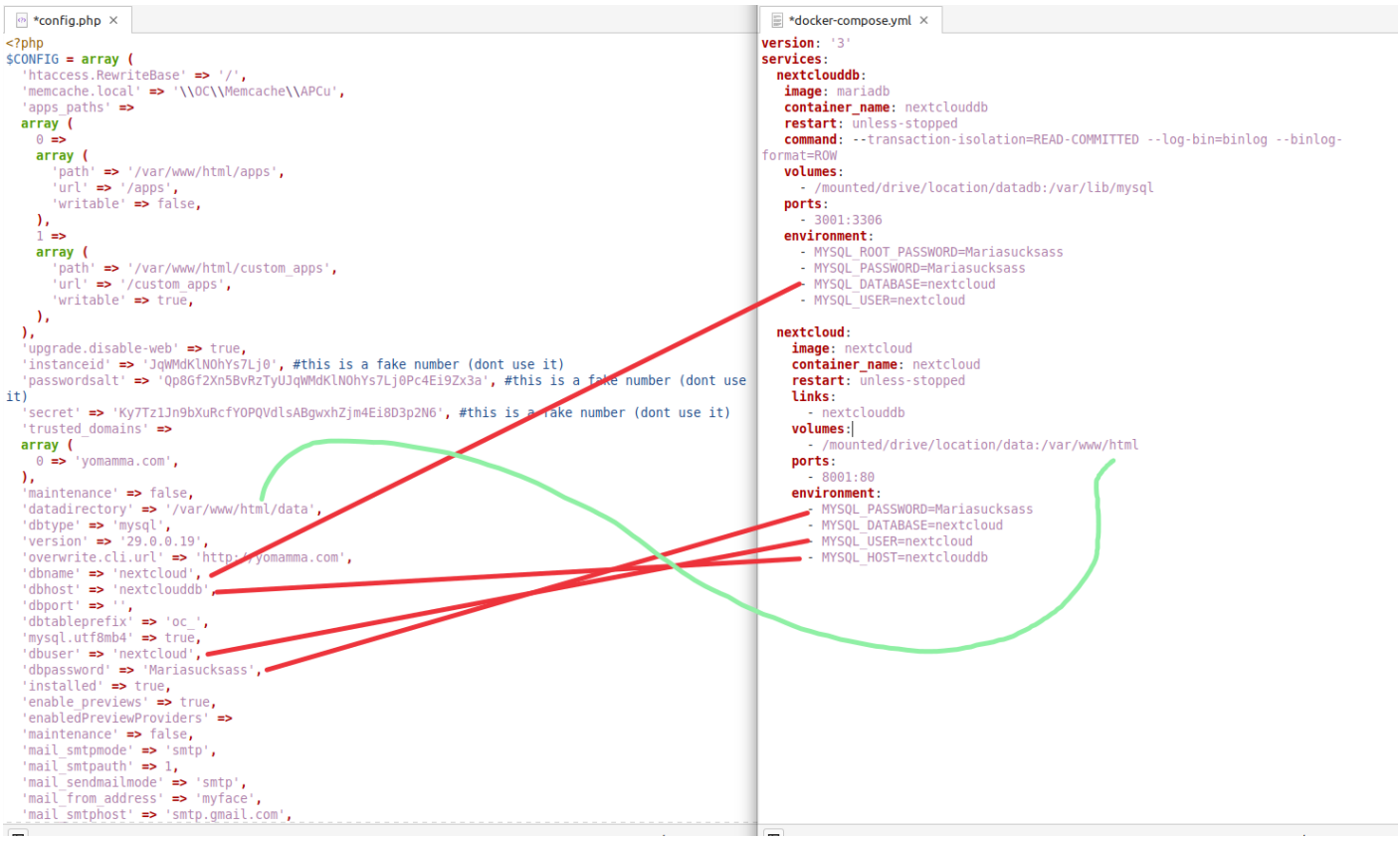
You can shut down your container.

Config file

So! Lets get into that config file and compare. Mine is located in `/home/server/nextcloud/data/config/`

I compare my .yaml file to the config.php, to make sure all my info is correct. Verify the other info not stated here like, trusted domains, overwrtie cli or anything else.

Note: the database info needs to match what is in your original config file so, if you need to, use that config file and copy the fields needed to make this new config file work.



Moving old data file

Now we can finally moving the main data in. Not all of it. ONLY the data folder that contains the users data. All files and folders that are before it, don't copy them over.

That means I'll be copying this one:

```
rsync -Aax /backup/directory/nextcloud/data/data/* /home/server/nextcloud/data/data
```

Notice the "data/data". We are deeper into the directory now. Here you should have all the files of all the users.

Start it up

If your new config file matches everything that is needed from your old one and, all your folders/files are copied over correctly, you should be good to go.

Start up a new container or... whatever.

If you came from an older Nextcloud version:

It might tell you to update using the terminal/command line. If so, try this.

If you dont need to update or, you finished the update, go to the next step.

Oh wait! there is no next step. You should only need to refresh your browser and it SHOULD work? Holy shit we did it!

This was my take on it. I just dont understand how nextcloud explains it. But...

If none of that works, try the below but I have not been able to get past step 5 using docker.

Setting up the New Server

1. **Install the Operating System**: Set up the new machine with the desired operating system.
2. **Configure Web Server and PHP**: Install and configure the web server (e.g., Apache or Nginx) and PHP for Nextcloud. Ensure that the PHP version matches Nextcloud's supported configuration, and all relevant PHP extensions are installed. Also, set the appropriate permissions and file upload size limits.
3. **Set up the Database**: Set up the database and ensure it's a Nextcloud-supported configuration.
4. **Copy Configuration (Optional)**: If your original machine was installed recently, you can safely copy the base configuration from the old server to the new server as a best practice.

Migrating Data from the Old Server

1. **Enable Maintenance Mode**: On the old server, stop Nextcloud and activate the maintenance mode. Wait for 6-7 minutes to allow all sync clients to register that the server is in maintenance mode.
    1. Note: if you do not have a running older system or have only files/folders of that system, there is no need to put it into maintenance mode as there is no front end client to pause.
2. **Stop Nextcloud**: After the waiting period, stop the application and/or web server that serves Nextcloud.
3. **Create a Database Backup**: Create a dump of the database on the old server and copy it to the new machine. Then, import the dump into the database on the new server. (See "Backup and Restoring backup" for more details.)
4. **Copy Nextcloud Files**: Copy all files from your Nextcloud instance on the old server, including the Nextcloud program files, data files, log files, and configuration files, to the new machine. Ensure that the data files retain their original timestamps (you can use `rsync` with the `-t` option) to prevent clients from re-downloading all files after the migration. Note that the file locations may vary depending on the original installation method and operating system. On the new system, make sure to place the files in the appropriate locations. If you change any paths, adapt the paths in the `config.php` file accordingly. (See "Backup and Restoring backup" for more details.)
5. **Check Data Fingerprint**: Check the `config.php` file of the old system to see if it has the `data-fingerprint` set to a non-empty value. If so, run the `maintenance:data-fingerprint` command on the new system, similar to how it's required when performing a backup restoration. (See "Restoring backup" for details.)

Testing and Finalizing the Migration

1. **Test the New Server**: While Nextcloud is still in maintenance mode on the new server (confirm!), start the database, web server, and application server. Access the migrated Nextcloud instance in your web browser. Confirm that you see the maintenance mode notice, log entries are written by both the web server and Nextcloud, and no error messages occur. Then, take Nextcloud out of maintenance mode and repeat the testing. Log in as an admin and confirm that Nextcloud is functioning normally.
2. **Update DNS**: Once you've confirmed that the new server is working correctly, change the CNAME entry in the DNS to point your users to the new server's location.

Full article from Nextcloud:

https://docs.nextcloud.com/server/stable/admin_manual/maintenance/migrating.html

Reset Bruteforce security | multiple invalid login attempts

Error Message:

We have detected multiple invalid login attempts from your IP. Therefore your next login is throttled up to 30 seconds

To fix it

You need to reset the bruteforce thingy.

If you are in a docker container, you can use this code from outside the container.

```
sudo docker exec -u 33 -it <docker-container-name> php occ security:bruteforce:reset <ip-address>
```

* remove the; <and everything inside it>

*In this method, the `-u 33` option is added to the `docker exec` command, which instructs Docker to run the command inside the container as the user with ID 33. (find what user number docker is using for yours if 33 doesn't work)

Speed up Nextcloud

Not uploading
Not uploading more than a few MB at a time
Not uploading more than a few files at a time
Not downloading
Downloading/Uploading slowly

Mess with these files:

Your php is probably messing with most of that. Find out where you php is located and also which version you have.
Min is located here
`/etc/php/8.1/fpm/php.ini`

Edit this file. Inside this here you should find a few fields:

```
upload_max_filesize = 16G
max_file_uploads = 3000
post_max_size = 16G
memory_limit = 512M
```

Edit them to a higher limit.

Restart php and server nginx/apache2

```
sudo systemctl restart php8.1-fpm.service
sudo systemctl restart apache2.service # For Apache
sudo systemctl restart nginx.service # For Nginx
```

Maintenance Mode for Nextcloud

Find where your Nextcloud config file is and edit it. Mine is stored here `/data/config/config.php`

In the last block of commands, add this somewhere in there.

```
  'maintenance' => true,
```

Save and quit. You should now be in Maintenance mode.

You don't always have to restart. It might just enter it right after saving.

Nextcloud update / Upgrade to newest version

To update Nextcloud when automatic updating is disabled in the config.php file from a local machine, you need to use the command line updater (occ upgrade). Here are the steps:

1. Open a terminal and navigate to the Nextcloud root directory.
2. Run the following command to start the command line updater:

```
sudo -u www-data php occ upgrade
```

To run the occ upgrade command from outside the Nextcloud Docker container named "nextcloud", you can use the docker exec command like this:

```
sudo docker exec --user www-data nextcloud php occ upgrade
```

Here's a breakdown of the command:

- `sudo docker exec` - This allows you to run a command inside a running Docker container.
- `--user www-data` - This specifies that the command should be run as the `www-data` user, which is the recommended user for running Nextcloud commands.
- `nextcloud` - This is the name of your Nextcloud container.
- `php occ upgrade` - This is the actual command to upgrade Nextcloud using the `occ` command-line tool.

So, the full command `sudo docker exec --user www-data nextcloud php occ upgrade` will execute the `php occ upgrade` command inside the "nextcloud" container as the `www-data` user, which is the recommended way to perform the Nextcloud upgrade.

Troubleshooting Nextcloud

trusted domain - Troubleshooting Nextcloud

Troubleshooting

- trusted domain
  - go to: nextcloud/config and edit the config.php
  - find: "trusted_domains" and add the domain you want to trust

```
'trusted_domains' =>
  array (
    0 => 'box.danicus.net',
    1 => '192.168.1.1',    (put in yourown ip address)
```

- **Note:** it's the domain you GO TO to access NextCloud...not where you're trying to access FROM.

Mobel app

Error message
  "no HTTP connection allowed"
Go to nextcloud config file, config.php and add
`'overwriteprotocol' => 'https', `

Upload failed request entity too large - Troubleshooting Nextcloud

"**Upload failed** request entity too large"

If you have a proxy, go into the config file and change the "client_max_body_size" line. If that line isn't there, add it under the http section.

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        client_max_body_size 1000M; #(right here)

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;
```

```
        include /etc/nginx/mime.types;
        default_type application/octet-stream;
```

I didn't have a line at all so I just put it in there.

Docker Container keeps restarting - Troubleshooting Nextcloud

This case, it was already working. Ended up pulling a new image but didnt know it.

First, check docker logs. Run the container and let it keep restarting so that it will give you an error log.

```
docker logs <nextcloud_container_name>
```

Error said:
Initializing nextcloud 31.0.1.2 ...
Can't start Nextcloud because upgrading from 29.0.0.19 to 31.0.1.2 is not supported.
It is only possible to upgrade one major version at a time. For example, if you want to upgrade from version 14 to 16, you will have to upgrade from version 14 to 15, then from 15 to 16.

That means files/data is for version 29 and i need to incrementally go up to V31.

Lets get the versions in between to install it correctly.

If the containers are running. stock them.

```
docker-compose down
```

Check Your Current Data

Your Nextcloud data (files and database) is tied to version 29.0.0.19. Ensure your volumes (./nextcloud and ./mariadb) are intact, as they contain your files and database. If you have backups, now’s a good time to confirm they’re ready—just in case.

Upgrade Step-by-Step

You’ll need to run intermediate versions of Nextcloud to perform the upgrades. Modify your docker-compose.yml to pull specific versions, starting with the next major version after 29.

Upgrade Step-by-Step

- You'll need to run intermediate versions of Nextcloud to perform the upgrades. Modify your docker-compose.yml to pull specific versions, starting with the next major version after 29.

```
services:
  nextcloud:
    image: nextcloud:30
    # ... rest of your config ...
  mariadb:
    # ... no change needed here ...
```

- Start the containers:

```
docker-compose up -d
```

- Check the logs to confirm it's running:

```
docker-compose logs -f
```

- Nextcloud should automatically run the upgrade process for version 30. Wait until it stabilizes (logs should stop showing errors and indicate a successful start).

Upgrade to 31.0.x

- Once version 30 is running fine, stop the containers:

```
docker-compose down
```

- Update docker-compose.yml to the desired version (31.0.1.2 or latest 31):

```
services:
  nextcloud:
    image: nextcloud:31
    # ... rest of your config ...
```

- Restart:

```
docker-compose up -d
```

- Check logs again:

```
docker-compose logs -f
```

The upgrade from 30 to 31 should proceed. Once complete, your Nextcloud should be fully operational on 31.0.1.2.

Error message : Update needed

Error message; (when loading nextcloud frontend)

> Update needed
> Please use the command line updater because updating via browser is disabled in your config.php.

After updating Nextcloud to new version, it gave me that error message.

Basically, you need to update the database to work with the new version. Here is how to do that.

. Access the Nextcloud Container

- Find your Nextcloud container name or ID:
  - Find your container name "docker ps"

Look for the container running nextcloud:30 (or whichever version).

- Access the container’s shell:

```
docker exec -it <nextcloud_container_name> bash
```

Run the Updater

- Inside the container, switch to the www-data user (Nextcloud’s default user): (this may not work. you may not need it anyway).

```
su - www-data
```

- Navigate to the Nextcloud directory (usually /var/www/html): (you might be there already)

```
cd /var/www/html
```

- Run the occ command to perform the update:

```
php occ upgrade
```

If that worked, great! your dont with the update.

The first time I did this, it did not work.
    You may need to take it off of maintenance mode in order to try it again.

Removing maintenance mode:

Inside the docker container, in the /var/www/html directory

```
php occ maintenance:mode --off
```

now try the update again.