nextcloud / **all-in-one** `Public`

📦 The official Nextcloud installation method. Provides easy deployment and maintenance with most features included in this one Nextcloud instance.

🔗 **nextcloud.com/blog/how-to-install-the-nextcloud-all-in-one-on-linux/**

⚖️ AGPL-3.0 license

⭐ **6.6k** stars   ⑂ **750** forks   ⑂ Branches   🏷️ Tags   ⌁ Activity

☆ Star	🔔 Notifications

<> **Code**   ⊙ Issues `21`   ⑂ Pull requests `35`   💬 Discussions   ▷ Actions   ⊘ Security   ⌁ Insights

⑂ main ⌄   ⑂ **31** Branches   🏷️ **204** Tags   ⑂   🏷️   🔍 Go to file   Go to file   <> **Code** ⌄   ⋯

👤 **szaimen** Update update-helm.sh: remove some default ⋯ ✓	1c4fa05 · 7 hours ago ⏱️

| 📁 .github	build(deps): bump shivammathur/setup-php in /….	yesterday
📁 Containers	nextcloud: update to 30.0.9	5 days ago
📁 app	remove `rel="noopener"` as it is implied by `targ…`	2 months ago
📁 community-containers	libretranslate-cc: add warning that it is deprecated	2 weeks ago
📁 manual-install	Yaml updates	3 weeks ago
📁 nextcloud-aio-helm-chart	Update update-helm.sh: remove some default	7 hours ago
📁 php	restore-instance test: try to fix timeout problem	5 days ago
📁 tests/QA	FTS: allow to adjust the ES_JAVA_OPTS	2 months ago
📄 .gitattributes	close #4491	last year
📄 .gitignore	Fix gitignore	9 months ago
📄 LICENSE	Add license file	4 years ago
📄 compose.yaml	docs: nextcloud/all-in-one moved to ghcr.io	3 weeks ago
📄 develop.md	move images to ghcr.io	3 weeks ago
📄 docker-ipv6-support.md	update detail	9 months ago
📄 docker-rootless.md	address review	6 months ago
📄 local-instance.md	some more adjustments	3 months ago
📄 manual-upgrade.md	docs: nextcloud/all-in-one moved to ghcr.io	3 weeks ago
📄 migration.md	migration.md: adjust `/path/to/nextcloud/data…`	3 months ago
📄 multiple-instances.md	docs: nextcloud/all-in-one moved to ghcr.io	3 weeks ago
📄 readme.md	Merge pull request #6218 from nextcloud/enh/6…	2 weeks ago
📄 reverse-proxy.md	rp-docs-apache: adjust comment	last week

Nextcloud All-in-One

The official Nextcloud installation method. Nextcloud AIO provides easy deployment and maintenance with most features included in this one Nextcloud instance.

Included are:

- Nextcloud
- High performance backend for Nextcloud Files
- Nextcloud Office (optional)
- High performance backend for Nextcloud Talk and TURN-server (optional)
- Nextcloud Talk Recording-server (optional)
- Backup solution (optional, based on BorgBackup)
- Imaginary (optional, for previews of heic, heif, illustrator, pdf, svg, tiff and webp)
- ClamAV (optional, Antivirus backend for Nextcloud)
- Fulltextsearch (optional)
- Whiteboard (optional)
- Docker Socket Proxy (optional, needed for Nextcloud App API)
- Community containers

▶ And much more:

Screenshots

First setup	After installation
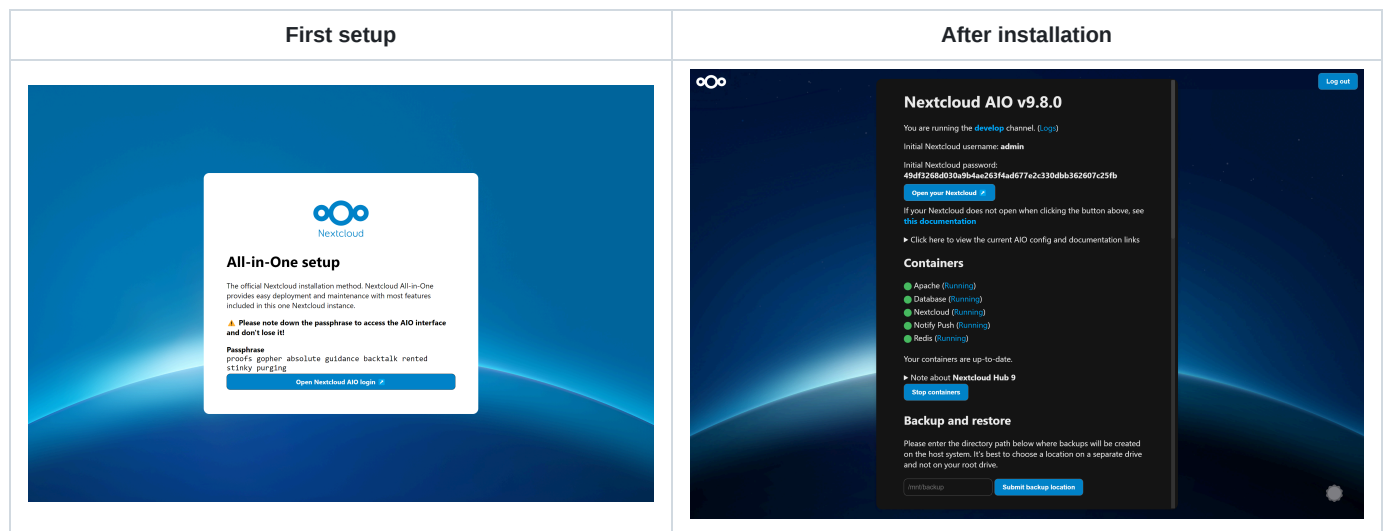
How to use this?

> ⓘ Note
>
> The following instructions are meant for installations without a web server or reverse proxy (like Apache, Nginx, Caddy, Cloudflare Tunnel and else) already being in place. If you want to run AIO behind a web server or reverse proxy (like Apache, Nginx, Caddy, Cloudflare Tunnel and else), see the reverse proxy documentation. Also, the instructions below are especially meant for Linux. For macOS see this, for Windows see this and for Synology see this.

1. Install Docker on your Linux installation by following the official documentation: https://docs.docker.com/engine/install/#supported-platforms.

> ⚠ Warning
>
> You could use the convenience script below to install docker. However we recommend to not blindly download and execute scripts as sudo. But if you feel like it, you can of course use it. See below:

▶ Using the convenience script

2. If you need ipv6 support, you should enable it by following https://github.com/nextcloud/all-in-one/blob/main/docker-ipv6-support.md.

3. Run the command below in order to start the container on Linux and without a web server or reverse proxy (like Apache, Nginx, Caddy, Cloudflare Tunnel and else) already in place:

```
# For Linux and without a web server or reverse proxy (like Apache, Nginx, Caddy, Cloudflare Tunnel and else)
already in place:
sudo docker run \
--init \
--sig-proxy=false \
--name nextcloud-aio-mastercontainer \
--restart always \
--publish 80:80 \
--publish 8080:8080 \
--publish 8443:8443 \
--volume nextcloud_aio_mastercontainer:/mnt/docker-aio-config \
--volume /var/run/docker.sock:/var/run/docker.sock:ro \
ghcr.io/nextcloud-releases/all-in-one:latest
```

▶ Explanation of the command

Note: You may be interested in adjusting Nextcloud's datadir to store the files in a different location than the default docker volume. See [this documentation](#) on how to do it.

4. After the initial startup, you should be able to open the Nextcloud AIO Interface now on port 8080 of this server.
   E.g. `https://ip.address.of.this.server:8080`
   ⚠️ **Important:** do always use an ip-address if you access this port and not a domain as HSTS might block access to it later! (It is also expected that this port uses a self-signed certificate due to security concerns which you need to accept in your browser)

   If your firewall/router has port 80 and 8443 open/forwarded and you point a domain to your server, you can get a valid certificate automatically by opening the Nextcloud AIO Interface via:
   `https://your-domain-that-points-to-this-server.tld:8443`

5. Please do not forget to open port `3478/TCP` and `3478/UDP` in your firewall/router for the Talk container!

FAQ

- [TOC](#)
  - [Where can I find additional documentation?](#)
  - [How does it work?](#)
  - [How to contribute?](#)
  - [How many users are possible?](#)
- [Network](#)
  - [Are reverse proxies supported?](#)
  - [Which ports are mandatory to be open in your firewall/router?](#)
  - [Explanation of used ports](#)
  - [Notes on Cloudflare (proxy/tunnel)](#)
  - [How to run Nextcloud behind a Cloudflare Tunnel?](#)
  - [How to run Nextcloud via Tailscale?](#)
  - [How to get Nextcloud running using the ACME DNS-challenge?](#)
  - [How to run Nextcloud locally? No domain wanted, or wanting intranet access within your LAN.](#)
  - [Can I use an ip-address for Nextcloud instead of a domain?](#)
  - [Can I run AIO offline or in an airgapped system?](#)
  - [Are self-signed certificates supported for Nextcloud?](#)
  - [Can I use AIO with multiple domains?](#)
  - [Are other ports than the default 443 for Nextcloud supported?](#)
  - [Can I run Nextcloud in a subdirectory on my domain?](#)
  - [How can I access Nextcloud locally?](#)
  - [How to skip the domain validation?](#)
  - [How to resolve firewall problems with Fedora Linux, RHEL OS, CentOS, SUSE Linux and others?](#)

- How to migrate from an already existing Nextcloud installation to Nextcloud AIO?
  - Backup
    - What is getting backed up by AIO's backup solution?
    - How to adjust borgs retention policy?
    - How to migrate from AIO to AIO?
    - Are remote borg backups supported?
    - Failure of the backup container in LXC containers
    - How to create the backup volume on Windows?
    - Pro-tip: Backup archives access
    - Delete backup archives manually
    - Sync local backups regularly to another drive
    - How to exclude Nextcloud's data directory or the preview folder from backup?
    - How to stop/start/update containers or trigger the daily backup from a script externally?
    - How to disable the backup section?
  - Addons
    - Fail2ban
    - LDAP
    - Netdata
    - USER_SQL
    - phpMyAdmin, Adminer or pgAdmin
    - Mail server
  - Miscellaneous
    - Requirements for integrating new containers
    - Update policy
    - How often are update notifications sent?
    - Huge docker logs

Where can I find additional documentation?

Some of the documentation is available on GitHub Discussions.

How does it work?

Nextcloud AIO is inspired by projects like Portainer that manage the docker daemon by talking to it through the docker socket directly. This concept allows a user to install only one container with a single command that does the heavy lifting of creating and managing all containers that are needed in order to provide a Nextcloud installation with most features included. It also makes updating a breeze and is not bound to the host system (and its slow updates) anymore as everything is in containers. Additionally, it is very easy to handle from a user perspective because a simple interface for managing your Nextcloud AIO installation is provided.

How to contribute?

See this issue for a list of feature requests that need help by contributors.

How many users are possible?

Up to 100 users are free, more are possible with Nextcloud Enterprise

Network

Are reverse proxies supported?

Yes. Please refer to the following documentation on this: reverse-proxy.md

Which ports are mandatory to be open in your firewall/router?

Only those (if you access the Mastercontainer Interface internally via port 8080):

- `443/TCP` for the Apache container

- `443/UDP` if you want to enable http3 for the Apache container
- `3478/TCP` and `3478/UDP` for the Talk container

Explanation of used ports

- `8080/TCP` : Mastercontainer Interface with self-signed certificate (works always, also if only access via IP-address is possible, e.g. `https://ip.address.of.this.server:8080/`) ⚠️ **Important:** do always use an ip-address if you access this port and not a domain as HSTS might block access to it later! (It is also expected that this port uses a self-signed certificate due to security concerns which you need to accept in your browser)
- `80/TCP` : redirects to Nextcloud (is used for getting the certificate via ACME http-challenge for the Mastercontainer)
- `8443/TCP` : Mastercontainer Interface with valid certificate (only works if port 80 and 8443 are open/forwarded in your firewall/router and you point a domain to your server. It generates a valid certificate then automatically and access via e.g. `https://public.domain.com:8443/` is possible.)
- `443/TCP` : will be used by the Apache container later on and needs to be open/forwarded in your firewall/router
- `443/UDP` : will be used by the Apache container later on and needs to be open/forwarded in your firewall/router if you want to enable http3
- `3478/TCP` and `3478/UDP` : will be used by the Turnserver inside the Talk container and needs to be open/forwarded in your firewall/router

Notes on Cloudflare (proxy/tunnel)

Since Cloudflare Proxy/Tunnel comes with a lot of limitations which are listed below, it is rather recommended to switch to Tailscale if possible.

- Cloudflare Proxy and Cloudflare Tunnel both require Cloudflare to perform TLS termination on their side and thus decrypt all the traffic on their infrastructure. This is a privacy concern and you will need to look for other solutions if it's unacceptable for you.
- Using Cloudflare Tunnel might potentially slow down Nextcloud since local access via the configured domain is not possible because TLS termination is in that case offloaded to Cloudflare's infrastructure. There is no way to disable this behavior in Cloudflare Tunnel.
- It is known that the domain validation may not work correctly behind Cloudflare since Cloudflare might block the validation attempt. You can simply skip it in that case by following: https://github.com/nextcloud/all-in-one#how-to-skip-the-domain-validation
- Make sure to disable Cloudflares Rocket Loader feature as otherwise Nextcloud's login prompt will not be shown.
- Cloudflare only supports uploading files up to 100 MB in the free plan, if you try to upload bigger files you will get an error (413 - Payload Too Large) if no chunking is used (e.g. for public uploads in the web, or if chunks are configured to be bigger than 100 MB in the clients or the web). If you need to upload bigger files, you need to disable the proxy option in your DNS settings. Note that this will both disable Cloudflare DDoS protection and Cloudflare Tunnel as these services require the proxy option to be enabled.
- If using Cloudflare Tunnel and the Nextcloud Desktop Client Set Chunking on Nextcloud Desktop Client
- Cloudflare only allows a max timeout of 100s for requests which is not configurable. This means that any server-side processing e.g. for assembling chunks for big files during upload that take longer than 100s will simply not work. See nextcloud/server#19223. If you need to upload big files reliably, you need to disable the proxy option in your DNS settings. Note that this will both disable Cloudflare DDoS protection and Cloudflare Tunnel as these services require the proxy option to be enabled.
- It is known that the in AIO included collabora (Nextcloud Office) does not work out of the box behind Cloudflare. To make it work, you need to add all Cloudflare IP-ranges to the wopi-allowlist in `https://yourdomain.com/settings/admin/richdocuments`
- Cloudflare Proxy might block the Turnserver for Nextcloud Talk from working correctly. You might want to disable Cloudflare Proxy thus. See #2463 (reply in thread)
- The built-in turn-server for Nextcloud Talk will not work behind Cloudflare Tunnel since it needs a separate port (by default 3478 or as chosen) available on the same domain. If you still want to use the feature, you will need to install your own turnserver or use a publicly available one and adjust and test your stun and turn settings in `https://yourdomain.com/settings/admin/talk` .
- If you get an error in Nextcloud's admin overview that the HSTS header is not set correctly, you might need to enable it in Cloudflare manually.
- If you are using AIO's built-in Reverse Proxy and don't use your own, then the certificate issuing may possibly not work out-of-the-box because Cloudflare might block the attempt. In that case you need to disable the Proxy feature at least temporarily in order to make it work. Note that this isn't an option if you need Cloudflare Tunnel as disabling the proxy would also disable Cloudflare Tunnel which would in turn make your server unreachable for the verification. See #1101.

How to run Nextcloud behind a Cloudflare Tunnel?

Although it does not seems like it is the case but from AIO perspective a Cloudflare Tunnel works like a reverse proxy. So please follow the reverse proxy documentation where is documented how to make it run behind a Cloudflare Tunnel. However please see the caveats before proceeding.

How to run Nextcloud via Tailscale?

For a reverse proxy example guide for Tailscale, see this guide by @flll: #5439

How to get Nextcloud running using the ACME DNS-challenge?

You can install AIO in reverse proxy mode where is also documented how to get it running using the ACME DNS-challenge for getting a valid certificate for AIO. See the reverse proxy documentation. (Meant is the `Caddy with ACME DNS-challenge` section). Also see https://github.com/dani-garcia/vaultwarden/wiki/Running-a-private-vaultwarden-instance-with-Let%27s-Encrypt-certs#getting-a-custom-caddy-build for additional docs on this topic.

How to run Nextcloud locally? No domain wanted, or wanting intranet access within your LAN.

If you do not want to open Nextcloud to the public internet, you may have a look at the following documentation on how to set it up locally: local-instance.md, but keep in mind you're still required to have https working properly.

Can I use an ip-address for Nextcloud instead of a domain?

No and it will not be added. If you only want to run it locally, you may have a look at the following documentation: local-instance.md. Recommended is to use Tailscale.

Can I run AIO offline or in an airgapped system?

No. This is not possible and will not be added due to multiple reasons: update checks, app installs via app-store, downloading additional docker images on demand and more.

Are self-signed certificates supported for Nextcloud?

No and they will not be. If you want to run it locally, without opening Nextcloud to the public internet, please have a look at the local instance documentation. Recommended is to use Tailscale.

Can I use AIO with multiple domains?

No and it will not be added. However you can use this feature in order to create multiple AIO instances, one for each domain.

Are other ports than the default 443 for Nextcloud supported?

No and they will not be. If port 443 and/or 80 is blocked for you, you may use Tailscale if you want to publish it online. If you already run a different service on port 443, please use a dedicated domain for Nextcloud and set it up correctly by following the reverse proxy documentation. However in all cases the Nextcloud interface will redirect you to port 443.

Can I run Nextcloud in a subdirectory on my domain?

No and it will not be added. Please use a dedicated (sub-)domain for Nextcloud and set it up correctly by following the reverse proxy documentation. Alternatively, you may use Tailscale if you want to publish it online.

How can I access Nextcloud locally?

Please note that local access is not possible if you are running AIO behind Cloudflare Tunnel since TLS proxying is in that case offloaded to Cloudflares infrastructure. You can fix this by setting up your own reverse proxy that handles TLS proxying locally and will make the steps below work.

Please make sure that if you are running AIO behind a reverse proxy, that the reverse proxy is configured to use port 443 on the server that runs it. Otherwise the steps below will not work.

Now that this is out of the way, the recommended way how to access Nextcloud locally, is to set up a local dns-server like a pi-hole and set up a custom dns-record for that domain that points to the internal ip-adddress of your server that runs Nextcloud AIO. Below are some guides:

- https://www.howtogeek.com/devops/how-to-run-your-own-dns-server-on-your-local-network/
- https://help.nextcloud.com/t/need-help-to-configure-internal-access/156075/6
- https://howchoo.com/pi/pi-hole-setup together with https://web.archive.org/web/20221203223505/https://docs.callitkarma.me/posts/PiHole-Local-DNS/
- https://dockerlabs.collabnix.com/intermediate/networking/Configuring_DNS.html Apart from that there is now a community container that can be added to the AIO stack: https://github.com/nextcloud/all-in-one/tree/main/community-containers/pi-hole

How to skip the domain validation?

If you are completely sure that you've configured everything correctly and are not able to pass the domain validation, you may skip the domain validation by adding `--env SKIP_DOMAIN_VALIDATION=true` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest` ! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used).

How to resolve firewall problems with Fedora Linux, RHEL OS, CentOS, SUSE Linux and others?

It is known that Linux distros that use [firewalld](#) as their firewall daemon have problems with docker networks. In case the containers are not able to communicate with each other, you may change your firewalld to use the iptables backend by running:

```
sudo sed -i 's/FirewallBackend=nftables/FirewallBackend=iptables/g' /etc/firewalld/firewalld.conf
sudo systemctl restart firewalld docker
```

Afterwards it should work.

See [https://dev.to/ozorest/fedora-32-how-to-solve-docker-internal-network-issue-22me](#) for more details on this. This limitation is even mentioned on the official firewalld website: [https://firewalld.org/#who-is-using-it](#)

What can I do to fix the internal or reserved ip-address error?

If you get an error during the domain validation which states that your ip-address is an internal or reserved ip-address, you can fix this by first making sure that your domain indeed has the correct public ip-address that points to the server and then adding `--add-host yourdomain.com:<public-ip-address>` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest` ! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) which will allow the domain validation to work correctly. And so that you know: even if the `A` record of your domain should change over time, this is no problem since the mastercontainer will not make any attempt to access the chosen domain after the initial domain validation.

Infrastructure

Which CPU architectures are supported?

You can check this on Linux by running: `uname -m`

- x86_64/x64/amd64
- aarch64/arm64/armv8

Disrecommended VPS providers

- *Older* Strato VPS using Virtuozzo caused problems though ones from Q3 2023 and later should work. If your VPS has a `/proc/user_beancounters` file and a low `numproc` limit set in it your server will likely misbehave once it reaches this limit which is very quickly reached by AIO, see [here](#).
- Hostingers VPS seem to miss a specific Kernel feature which is required for AIO to run correctly. See [here](#).

Recommended VPS

In general recommended VPS are those that are KVM/non-virtualized as Docker should work best on them.

Note on storage options

- SD-cards are disrecommended for AIO since they cripple the performance and they are not meant for many write operations which is needed for the database and other parts
- SSD storage is recommended
- HDD storage should work as well but is of course much slower than SSD storage

Are there known problems when SELinux is enabled?

Yes. If SELinux is enabled, you might need to add the `--security-opt label:disable` option to the docker run command of the mastercontainer in order to allow it to access the docker socket (or `security_opt: ["label:disable"]` in compose.yaml). See [#485](#)

Customization

How to change the default location of Nextcloud's Datadir?

> ⚠️ Warning
>
> Do not set or adjust this value after the initial Nextcloud installation is done! If you still want to do it afterwards, see [this](#) on how to do it.

You can configure the Nextcloud container to use a specific directory on your host as data directory. You can do so by adding the environmental variable `NEXTCLOUD_DATADIR` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used). Allowed values for that variable are strings that start with `/` and are not equal to `/`. The chosen directory or volume will then be mounted to `/mnt/ncdata` inside the container.

- An example for Linux is `--env NEXTCLOUD_DATADIR="/mnt/ncdata"`. ⚠️ Please note: If you should be using an external BTRFS drive that is mounted to `/mnt/ncdata`, make sure to choose a subfolder like e.g. `/mnt/ncdata/nextcloud` as datadir, since the root folder is not suited as datadir in that case. See [#2696](#).
- On macOS it might be `--env NEXTCLOUD_DATADIR="/var/nextcloud-data"`
- For Synology it may be `--env NEXTCLOUD_DATADIR="/volume1/docker/nextcloud/data"`.
- On Windows it might be `--env NEXTCLOUD_DATADIR="/run/desktop/mnt/host/c/ncdata"`. (This path is equivalent to `C:\ncdata` on your Windows host so you need to translate the path accordingly. Hint: the path that you enter needs to start with `/run/desktop/mnt/host/`. Append to that the exact location on your windows host, e.g. `c/ncdata` which is equivalent to `C:\ncdata`.) ⚠️ **Please note**: This does not work with external drives like USB or network drives and only with internal drives like SATA or NVME drives.
- Another option is to provide a specific volume name here with: `--env NEXTCLOUD_DATADIR="nextcloud_aio_nextcloud_datadir"`. This volume needs to be created beforehand manually by you in order to be able to use it. e.g. on Windows with:

  ```
  docker volume create ^
  --driver local ^
  --name nextcloud_aio_nextcloud_datadir ^
  -o device="/host_mnt/e/your/data/path" ^
  -o type="none" ^
  -o o="bind"
  ```

  In this example, it would mount `E:\your\data\path` into the volume so for a different location you need to adjust `/host_mnt/e/your/data/path` accordingly.

How to store the files/installation on a separate drive?

You can move the whole docker library and all its files including all Nextcloud AIO files and folders to a separate drive by first mounting the drive in the host OS (NTFS is not supported and ext4 is recommended as FS) and then following this tutorial:
https://www.guguweb.com/2019/02/07/how-to-move-docker-data-directory-to-another-location-on-ubuntu/
(Of course docker needs to be installed first for this to work.)

⚠️ If you encounter errors from richdocuments in your Nextcloud logs, check in your Collabora container if the message "Capabilities are not set for the coolforkit program." appears. If so, follow these steps:

1. Stop all the containers from the AIO Interface.
2. Go to your terminal and delete the Collabora container (`docker rm nextcloud-aio-collabora`) AND the Collabora image (`docker image rm nextcloud/aio-collabora`).
3. You might also want to prune your Docker (`docker system prune`) (no data will be lost).
4. Restart your containers from the AIO Interface.

This should solve the problem.

How to allow the Nextcloud container to access directories on the host?

By default, the Nextcloud container is confined and cannot access directories on the host OS. You might want to change this when you are planning to use local external storage in Nextcloud to store some files outside the data directory and can do so by adding the environmental variable `NEXTCLOUD_MOUNT` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used). Allowed values for that variable are strings that start with `/` and are not equal to `/`.

- Two examples for Linux are `--env NEXTCLOUD_MOUNT="/mnt/"` and `--env NEXTCLOUD_MOUNT="/media/"`.
- On macOS it might be `--env NEXTCLOUD_MOUNT="/Volumes/your_drive/"`
- For Synology it may be `--env NEXTCLOUD_MOUNT="/volume1/"`.
- On Windows it might be `--env NEXTCLOUD_MOUNT="/run/desktop/mnt/host/d/your-folder/"`. (This path is equivalent to `D:\your-folder` on your Windows host so you need to translate the path accordingly. Hint: the path that you enter needs to start with `/run/desktop/mnt/host/`. Append to that the exact location on your windows host, e.g. `d/your-folder/` which is equivalent to `D:\your-folder`.) ⚠️ **Please note**: This does not work with external drives like USB or network drives and only with internal drives like SATA or NVME drives.

After using this option, please make sure to apply the correct permissions to the directories that you want to use in Nextcloud. E.g. `sudo chown -R 33:0 /mnt/your-drive-mountpoint` and `sudo chmod -R 750 /mnt/your-drive-mountpoint` should make it work on Linux when you have used `--env NEXTCLOUD_MOUNT="/mnt/"`. On Windows you could do this e.g. with `docker exec -it nextcloud-aio-nextcloud chown -R 33:0 /run/desktop/mnt/host/d/your-folder/` and `docker exec -it nextcloud-aio-nextcloud chmod -R 750 /run/desktop/mnt/host/d/your-folder/`.

You can then navigate to `https://your-nc-domain.com/settings/apps/disabled`, activate the external storage app, navigate to `https://your-nc-domain.com/settings/admin/externalstorages` and add a local external storage directory that will be accessible inside the container at the same place that you've entered. E.g. `/mnt/your-drive-mountpoint` will be mounted to `/mnt/your-drive-mountpoint` inside the container, etc.

Be aware though that these locations will not be covered by the built-in backup solution - but you can add further Docker volumes and host paths that you want to back up after the initial backup is done.

> ⓘ Note
>
> If you can't see the type "local storage" in the external storage admin options, a restart of the containers from the AIO interface may be required.

How to adjust the Talk port?

By default will the talk container use port `3478/UDP` and `3478/TCP` for connections. This should be set to something higher than 1024! You can adjust the port by adding e.g. `--env TALK_PORT=3478` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and adjusting the port to your desired value. Best is to use a port over 1024, so e.g. 3479 to not run into this: [#2517](#)

How to adjust the upload limit for Nextcloud?

By default, public uploads to Nextcloud are limited to a max of 16G (logged in users can upload much bigger files using the webinterface or the mobile/desktop clients, since chunking is used in that case). You can adjust the upload limit by providing `--env NEXTCLOUD_UPLOAD_LIMIT=16G` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must start with a number and end with `G` e.g. `16G`.

How to adjust the max execution time for Nextcloud?

By default, uploads to Nextcloud are limited to a max of 3600s. You can adjust the upload time limit by providing `--env NEXTCLOUD_MAX_TIME=3600` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must be a number e.g. `3600`.

How to adjust the PHP memory limit for Nextcloud?

By default, each PHP process in the Nextcloud container is limited to a max of 512 MB. You can adjust the memory limit by providing `--env NEXTCLOUD_MEMORY_LIMIT=512M` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must start with a number and end with `M` e.g. `1024M`.

How to change the Nextcloud apps that are installed on the first startup?

You might want to adjust the Nextcloud apps that are installed upon the first startup of the Nextcloud container. You can do so by adding `--env NEXTCLOUD_STARTUP_APPS="deck twofactor_totp tasks calendar contacts notes"` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must be a string with small letters a-z, 0-9, spaces and hyphens or '_'. You can disable shipped and by default enabled apps by adding a hyphen in front of the appid. E.g. `-contactsinteraction`.

How to add OS packages permanently to the Nextcloud container?

Some Nextcloud apps require additional external dependencies that must be bundled within Nextcloud container in order to work correctly. As we cannot put each and every dependency for all apps into the container - as this would make the project quickly unmaintainable - there is an official way in which you can add additional dependencies into the Nextcloud container. However note that doing this is disrecommended since we do not test Nextcloud apps that require external dependencies.

You can do so by adding `--env NEXTCLOUD_ADDITIONAL_APKS="imagemagick dependency2 dependency3"` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must be a string with small letters a-z, digits 0-9, spaces, dots and hyphens or '_'. You can find available packages here: https://pkgs.alpinelinux.org/packages?branch=v3.21. By default `imagemagick` is added. If you want to keep it, you need to specify it as well.

How to add PHP extensions permanently to the Nextcloud container?

Some Nextcloud apps require additional php extensions that must be bundled within Nextcloud container in order to work correctly. As we cannot put each and every dependency for all apps into the container - as this would make the project quickly unmaintainable - there is an official way in which you can add additional php extensions into the Nextcloud container. However note that doing this is disrecommended since we do not test Nextcloud apps that require additional php extensions.

You can do so by adding `--env NEXTCLOUD_ADDITIONAL_PHP_EXTENSIONS="imagick extension1 extension2"` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. It must be a string with small letters a-z, digits 0-9, spaces, dots and hyphens or '_'. You can find available extensions here: https://pecl.php.net/packages.php. By default `imagick` is added. If you want to keep it, you need to specify it as well.

What about the pdlib PHP extension for the facerecognition app?

The facerecognition app requires the pdlib PHP extension to be installed. Unfortunately, it is not available on PECL nor via PHP core, so there is no way to add this into AIO currently. However you can use this community container in order to run facerecognition.

How to enable hardware acceleration for Nextcloud?

Some container can use GPU acceleration to increase performance like memories app allows to enable hardware transcoding for videos.

With open source drivers MESA for AMD, Intel and new drivers `Nouveau` for Nvidia

> ⚠ Warning
>
> This only works if the `/dev/dri` device is present on the host! If it does not exist on your host, don't proceed as otherwise the Nextcloud container will fail to start! If you are unsure about this, better do not proceed with the instructions below. Make sure that your driver is correctly configured on the host.

A list of supported device can be fond in MESA 3D documentation.

This method use the Direct Rendering Infrastructure with the access to the `/dev/dri` device.

In order to use that, you need to add `--env NEXTCLOUD_ENABLE_DRI_DEVICE=true` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) which will mount the `/dev/dri` device into the container.

With proprietary drivers for Nvidia ⚠ BETA

> ⚠ Warning

This only works if the Nvidia Toolkit is installed on the host and an NVIDIA GPU is enabled! Make sure that it is correctly configured on the host. If it does not exist on your host, don't proceed as otherwise the Nextcloud container will fail to start! If you are unsure about this, better do not proceed with the instructions below.

This feature is in beta. Since the proprietary, we haven't a lot of user using proprietary drivers, we can't guarantee the stability of this feature. Your feedback is welcome.

This method use the [Nvidia Container Toolkit](#) with the nvidia runtime.

In order to use that, you need to add `--env NEXTCLOUD_ENABLE_NVIDIA_GPU=true` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) which will enable the nvidia runtime.

If you're using WSL2 and want to use the NVIDIA runtime, please follow the instructions to [install the NVIDIA Container Toolkit meta-version in WSL](#).

How to keep disabled apps?

In certain situations you might want to keep Nextcloud apps that are disabled in the AIO interface and not uninstall them if they should be installed in Nextcloud. You can do so by adding `--env NEXTCLOUD_KEEP_DISABLED_APPS=true` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used).

> ⚠️ Warning
>
> Doing this might cause unintended problems in Nextcloud if an app that requires an external dependency is still installed but the external dependency not for example.

How to trust user-defined Certification Authorities (CA)?

> ⓘ Note
>
> Please note, that this feature is only intended to make LDAPS connections with self-signed certificates work. It will not make other interconnectivity between the different containers work, as they expect a valid publicly trusted certificate like one from Let's Encrypt.

For some applications it might be necessary to establish a secure connection to another host/server which is using a certificate issued by a Certification Authority that is not trusted out of the box. An example could be configuring LDAPS against a domain controller (Active Directory or Samba-based) of an organization.

You can make the Nextcloud container trust any Certification Authority by providing the environmental variable `NEXTCLOUD_TRUSTED_CACERTS_DIR` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used). The value of the variables should be set to the absolute paths of the directory on the host, which contains one or more Certification Authorities certificates. You should use X.509 certificates, Base64 encoded. (Other formats may work but have not been tested!) All the certificates in the directory will be trusted.

When using `docker run`, the environmental variable can be set with `--env NEXTCLOUD_TRUSTED_CACERTS_DIR=/path/to/my/cacerts`.

In order for the value to be valid, the path should start with `/` and not end with `/` and point to an existing **directory**. Pointing the variable directly to a certificate **file** will not work and may also break things.

How to disable Collabora's Seccomp feature?

The Collabora container enables Seccomp by default, which is a security feature of the Linux kernel. On systems without this kernel feature enabled, you need to provide `--env COLLABORA_SECCOMP_DISABLED=true` to the initial docker run command in order to make it work. If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used.

How to adjust the Fulltextsearch Java options?

The Fulltextsearch Java options are by default set to `-Xms512M -Xmx512M` which might not be enough on some systems. You can adjust this by adding e.g. `--env FULLTEXTSEARCH_JAVA_OPTIONS="-Xms1024M -Xmx1024M"` to the initial docker run command. If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used.

Guides

How to run AIO on macOS?

On macOS, there is only one thing different in comparison to Linux: instead of using `--volume` `/var/run/docker.sock:/var/run/docker.sock:ro` , you need to use `--volume /var/run/docker.sock.raw:/var/run/docker.sock:ro` to run it after you installed Docker Desktop (and don't forget to enable ipv6 if you should need that). Apart from that it should work and behave the same like on Linux.

Also, you may be interested in adjusting Nextcloud's Datadir to store the files on the host system. See this documentation on how to do it.

How to run AIO on Windows?

On Windows, install Docker Desktop (and don't forget to enable ipv6 if you should need that) and run the following command in the command prompt:

```
docker run ^
--init ^
--sig-proxy=false ^
--name nextcloud-aio-mastercontainer ^
--restart always ^
--publish 80:80 ^
--publish 8080:8080 ^
--publish 8443:8443 ^
--volume nextcloud_aio_mastercontainer:/mnt/docker-aio-config ^
--volume //var/run/docker.sock:/var/run/docker.sock:ro ^
ghcr.io/nextcloud-releases/all-in-one:latest
```

Also, you may be interested in adjusting Nextcloud's Datadir to store the files on the host system. See this documentation on how to do it.

> ⓘ Note
>
> Almost all commands in this project's documentation use `sudo docker ...` . Since `sudo` is not available on Windows, you simply remove `sudo` from the commands and they should work.

How to run AIO on Synology DSM

On Synology, there are two things different in comparison to Linux: instead of using `--volume` `/var/run/docker.sock:/var/run/docker.sock:ro` , you need to use `--volume /volume1/docker/docker.sock:/var/run/docker.sock:ro` to run it. You also need to add `--env WATCHTOWER_DOCKER_SOCKET_PATH="/volume1/docker/docker.sock"` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`). Apart from that it should work and behave the same like on Linux. Obviously the Synology Docker GUI will not work with that so you will need to either use SSH or create a user-defined script task in the task scheduler as the user 'root' in order to run the command.

> ⓘ Note
>
> It is possible that the docker socket on your Synology is located in `/var/run/docker.sock` like the default on Linux. Then you can just use the Linux command without having to change anything - you will notice this when you try to start the container and it says that the bind mount failed. E.g. `docker: Error response from daemon: Bind mount failed: '/volume1/docker/docker.sock' does not exists.`

Also, you may be interested in adjusting Nextcloud's Datadir to store the files on the host system. See this documentation on how to do it.

You'll also need to adjust Synology's firewall, see below:

▶ Click here to expand

How to run AIO with Portainer?

The easiest way to run it with Portainer on Linux is to use Portainer's stacks feature and use this docker-compose file in order to start AIO correctly.

Can I run AIO on TrueNAS SCALE?

With the Truenas Scale Release 24.10.0 (which was officially released on October 29th 2024 as a stable release) IX Systems ditched the Kubernetes integration and implemented a fully working docker environment.

For a more complete guide, see this guide by @zybster: [#5506](#)

On older TrueNAS SCALE releases with Kubernetes environment, there are two ways to run AIO. The preferred one is to run AIO inside a VM. This is necessary since they do not expose the docker socket for containers on the host, you also cannot use docker-compose on it thus and it is also not possible to run custom helm-charts that are not explicitly written for TrueNAS SCALE.

Another but untested way is to install Portainer on your TrueNAS SCALE from here https://truecharts.org/charts/stable/portainer/installation-notes and add the Helm-chart repository https://nextcloud.github.io/all-in-one/ into Portainer by following https://docs.portainer.io/user/kubernetes/helm. More docs on AIOs Helm Chart are available here: https://github.com/nextcloud/all-in-one/tree/main/nextcloud-aio-helm-chart#nextcloud-aio-helm-chart.

How to run `occ` commands?

Simply run the following: `sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ your-command`. Of course `your-command` needs to be exchanged with the command that you want to run.

How to resolve `Security & setup warnings displays the "missing default phone region" after initial install`?

Simply run the following command: `sudo docker exec --user www-data nextcloud-aio-nextcloud php occ config:system:set default_phone_region --value="yourvalue"`. Of course you need to modify `yourvalue` based on your location. Examples are `DE`, `US` and `GB`. See this list for more codes: https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2#Officially_assigned_code_elements

How to run multiple AIO instances on one server?

See [multiple-instances.md](#) for some documentation on this.

Bruteforce protection FAQ

Nextcloud features a built-in bruteforce protection which may get triggered and will block an ip-address or disable a user. You can unblock an ip-address by running `sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ security:bruteforce:reset <ip-address>` and enable a disabled user by running `sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ user:enable <name of user>`. See https://docs.nextcloud.com/server/latest/admin_manual/configuration_server/occ_command.html#security for further information.

How to switch the channel?

You can switch to a different channel like e.g. the beta channel or from the beta channel back to the latest channel by stopping the mastercontainer, removing it (no data will be lost) and recreating the container using the same command that you used initially to create the mastercontainer. You simply need to change the last line `ghcr.io/nextcloud-releases/all-in-one:latest` to `ghcr.io/nextcloud-releases/all-in-one:beta` and vice versa.

How to update the containers?

If we push new containers to `latest`, you will see in the AIO interface below the `containers` section that new container updates were found. In this case, just press `Stop containers` and `Start and update containers` in order to update the containers. The mastercontainer has its own update procedure though. See below. And don't forget to back up the current state of your instance using the built-in backup solution before starting the containers again! Otherwise you won't be able to restore your instance easily if something should break during the update.

If a new `mastercontainer` update was found, you'll see a note below the `Stop containers` button that allows to show the changelog. If you click that button and the containers are stopped, you will see a new button that allows to update the mastercontainer. After doing so and after the update is gone through, you will have the option again to `Start and update containers`. It is recommended to create a backup before clicking the `Start and update containers` button.

Additionally, there is a cronjob that runs once a day that checks for container and mastercontainer updates and sends a notification to all Nextcloud admins if a new update was found.

How to easily log in to the AIO interface?

If your Nextcloud is running and you are logged in as admin in your Nextcloud, you can easily log in to the AIO interface by opening `https://yourdomain.tld/settings/admin/overview` which will show a button on top that enables you to log in to the AIO interface by just clicking on this button.

> ⓘ **Note**
>
> You can change the domain/ip-address/port of the button by simply stopping the containers, visiting the AIO interface from the correct and desired domain/ip-address/port and clicking once on `Start containers`.

How to change the domain?

> ⓘ **Note**
>
> Editing the configuration.json manually and making a mistake may break your instance so please create a backup first!

If you set up a new AIO instance, you need to enter a domain. Currently there is no way to change this domain afterwards from the AIO interface. So in order to change it, you need to edit the configuration.json manually using `sudo docker run -it --rm --volume nextcloud_aio_mastercontainer:/mnt/docker-aio-config:rw alpine sh -c "apk add --no-cache nano && nano /mnt/docker-aio-config/data/configuration.json"`, substitute each occurrence of your old domain with your new domain and save and write out the file. Afterwards restart your containers from the AIO interface and everything should work as expected if the new domain is correctly configured. If you are running AIO behind a web server or reverse proxy (like Apache, Nginx, Caddy, Cloudflare Tunnel and else), you need to obviously also change the domain in your reverse proxy config.

Additionally, after restarting the containers, you need to open the admin settings and update some values manually that cannot be changed automatically. Here is a list of some known places:

- `https://your-nc-domain.com/settings/admin/talk` for Turn/Stun server and Signaling Server if you enabled Talk via the AIO interface
- `https://your-nc-domain.com/settings/admin/theming` for the theming URL
- `https://your-nc-domain.com/settings/admin/app_api` for the deploy daemon if you enabled the App API via the AIO interface

How to properly reset the instance?

If something goes unexpected routes during the initial installation, you might want to reset the AIO installation to be able to start from scratch.

> ⓘ **Note**
>
> If you already have it running and have data on your instance, you should not follow these instructions as it will delete all data that is coupled to your AIO instance.

Here is how to reset the AIO instance properly:

1. Stop all containers if they are running from the AIO interface
2. Stop the mastercontainer with `sudo docker stop nextcloud-aio-mastercontainer`
3. If the domaincheck container is still running, stop it with `sudo docker stop nextcloud-aio-domaincheck`
4. Check that no AIO containers are running anymore by running `sudo docker ps --format {{.Names}}`. If no `nextcloud-aio` containers are listed, you can proceed with the steps below. If there should be some, you will need to stop them with `sudo docker stop <container_name>` until no one is listed anymore.
5. Check which containers are stopped: `sudo docker ps --filter "status=exited"`
6. Now remove all these stopped containers with `sudo docker container prune`
7. Delete the docker network with `sudo docker network rm nextcloud-aio`
8. Check which volumes are dangling with `sudo docker volume ls --filter "dangling=true"`
9. Now remove all these dangling volumes: `sudo docker volume prune --filter all=1` (on Windows you might need to remove some volumes afterwards manually with `docker volume rm nextcloud_aio_backupdir`, `docker volume rm nextcloud_aio_nextcloud_datadir`).
10. If you've configured `NEXTCLOUD_DATADIR` to a path on your host instead of the default volume, you need to clean that up as well. (E.g. by simply deleting the directory).
11. Make sure that no volumes are remaining with `sudo docker volume ls --format {{.Name}}`. If no `nextcloud-aio` volumes are listed, you can proceed with the steps below. If there should be some, you will need to remove them with `sudo docker volume rm <volume_name>` until no one is listed anymore.
12. Optional: You can remove all docker images with `sudo docker image prune -a`.

13. And you are done! Now feel free to start over with the recommended docker run command!

Can I use a CIFS/SMB share as Nextcloud's datadir?

Sure. Add this to the `/etc/fstab` file on the host system:

`<your-storage-host-and-subpath> <your-mount-dir> cifs rw,mfsymlinks,seal,credentials=<your-credentials-file>,uid=33,gid=0,file_mode=0770,dir_mode=0770 0 0`

(Of course you need to modify `<your-storage-host-and-subpath>` , `<your-mount-dir>` and `<your-credentials-file>` for your specific case.)

One example could look like this:

`//your-storage-host/subpath /mnt/storagebox cifs rw,mfsymlinks,seal,credentials=/etc/storage-credentials,uid=33,gid=0,file_mode=0770,dir_mode=0770 0 0`

and add into `/etc/storage-credentials` :

```
username=<smb/cifs username>
password=<password>
```

(Of course you need to modify `<smb/cifs username>` and `<password>` for your specific case.)

Now you can use `/mnt/storagebox` as Nextcloud's datadir like described in the section above this one.

Can I run this with Docker swarm?

Yes. For that to work, you need to use and follow the [manual-install documentation](#).

📖 **README**   ⚖️ AGPL-3.0 license   ⚖️ Security   ☰

Yes. For that to work, you need to use and follow the [helm-chart documentation](#).

How to run this with Docker rootless?

You can run AIO also with docker rootless. How to do this is documented here: [docker-rootless.md](#)

Can I run this with Podman instead of Docker?

Since Podman is not 100% compatible with the Docker API, Podman is not supported (since that would add yet another platform where the maintainer would need to test on). However you can use and follow the [manual-install documentation](#) to get AIO's containers running with Podman or use Docker rootless, as described in the above section. Also there is this now: [#3487](#)

Access/Edit Nextcloud files/folders manually

The files and folders that you add to Nextcloud are by default stored in the following docker directory: `nextcloud_aio_nextcloud:/mnt/ncdata/` (usually `/var/lib/docker/volumes/nextcloud_aio_nextcloud_data/_data/` on linux host systems). If needed, you can modify/add/delete files/folders there but **ATTENTION**: be very careful when doing so because you might corrupt your AIO installation! Best is to create a backup using the built-in backup solution before editing/changing files/folders in there because you will then be able to restore your instance to the backed up state.

After you are done modifying/adding/deleting files/folders, don't forget to apply the correct permissions by running: `sudo docker exec nextcloud-aio-nextcloud chown -R 33:0 /mnt/ncdata/` and `sudo docker exec nextcloud-aio-nextcloud chmod -R 750 /mnt/ncdata/` and rescan the files with `sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ files:scan --all` .

How to edit Nextclouds config.php file with a texteditor?

You can edit Nextclouds config.php file directly from the host with your favorite text editor. E.g. like this: `sudo docker run -it --rm --volume nextcloud_aio_nextcloud:/var/www/html:rw alpine sh -c "apk add --no-cache nano && nano /var/www/html/config/config.php"` . Make sure to not break the file though which might corrupt your Nextcloud instance otherwise. In best case, create a backup using the built-in backup solution before editing the file.

How to change default files by creating a custom skeleton directory?

All users see a set of [default files and folders](#) as dictated by Nextcloud's configuration. To change these default files and folders a custom skeleton directory must first be created; this can be accomplished by copying your skeleton files `sudo docker cp --follow-link /path/to/nextcloud/skeleton/ nextcloud-aio-nextcloud:/mnt/ncdata/skeleton/` , applying the correct permissions with `sudo docker exec nextcloud-aio-nextcloud chown -R 33:0 /mnt/ncdata/skeleton/` and `sudo docker exec nextcloud-aio-nextcloud chmod -R 750 /mnt/ncdata/skeleton/` and setting the skeleton directory option with `sudo docker exec --user www-data -it nextcloud-aio-nextcloud php occ config:system:set skeletondirectory --value="/mnt/ncdata/skeleton"` . Further information is available in the Nextcloud documentation on [configuration parameters for the skeleton directory](#).

How to adjust the version retention policy and trashbin retention policy?

By default, AIO sets the `versions_retention_obligation` and `versions_retention_obligation` both to `auto, 30` which means that versions and items in the trashbin get deleted after 30 days. If you want to change this, see https://docs.nextcloud.com/server/latest/admin_manual/configuration_files/file_versioning.html.

How to enable automatic updates without creating a backup beforehand?

If you have an external backup solution, you might want to enable automatic updates without creating a backup first. However note that doing this is disrecommended since you will not be able to easily create and restore a backup from the AIO interface anymore and you need to make sure to shut down all the containers properly before creating the backup, e.g. by stopping them from the AIO interface first.

But anyhow, is here a guide that helps you automate the whole procedure:

▶ Click here to expand

You can simply copy and paste the script into a file e.g. named `shutdown-script.sh` e.g. here: `/root/shutdown-script.sh` .

Afterwards apply the correct permissions with `sudo chown root:root /root/shutdown-script.sh` and `sudo chmod 700 /root/shutdown-script.sh` . Then you can create a cronjob that runs it on a schedule e.g. runs the script at `04:00` each day like this:

1. Open the cronjob with `sudo crontab -u root -e` (and choose your editor of choice if not already done. I'd recommend nano).
2. Add the following new line to the crontab if not already present: `0 4 * * * /root/shutdown-script.sh` which will run the script at 04:00 each day.
3. save and close the crontab (when using nano the shortcuts for this are `Ctrl + o` and then `Enter` to save, and close the editor with `Ctrl + x`).

After that is in place, you should schedule a backup from your backup solution that creates a backup after AIO is shut down properly. Hint: If your backup runs on the same host, make sure to at least back up all docker volumes and additionally Nextcloud's datadir if it is not stored in a docker volume.

Afterwards, you can create a second script that automatically updates the containers:

▶ Click here to expand

You can simply copy and paste the script into a file e.g. named `automatic-updates.sh` e.g. here: `/root/automatic-updates.sh` .

Afterwards apply the correct permissions with `sudo chown root:root /root/automatic-updates.sh` and `sudo chmod 700 /root/automatic-updates.sh` . Then you can create a cronjob that runs e.g. at `05:00` each day like this:

1. Open the cronjob with `sudo crontab -u root -e` (and choose your editor of choice if not already done. I'd recommend nano).
2. Add the following new line to the crontab if not already present: `0 5 * * * /root/automatic-updates.sh` which will run the script at 05:00 each day.
3. save and close the crontab (when using nano the shortcuts for this are `Ctrl + o` then `Enter` to save, and close the editor with `Ctrl + x`).

Securing the AIO interface from unauthorized ACME challenges

[By design](#), Caddy that runs inside the mastercontainer, which handles automatic TLS certificate generation for the AIO interface, is vulnerable to receiving DNS challenges for arbitrary hostnames from anyone on the internet. While this does not compromise your server's security, it can result in cluttered logs and rejected certificate renewal attempts due to rate limit abuse. To mitigate this issue, it is recommended to place the AIO interface behind a VPN and/or limit its public exposure.

How to migrate from an already existing Nextcloud installation to Nextcloud AIO?

Please see the following documentation on this: [migration.md](#)

Backup

Nextcloud AIO provides a backup solution based on [BorgBackup](#). These backups act as a restore point in case the installation gets corrupted. By using this tool, backups are incremental, differential, compressed and encrypted – so only the first backup will take a while. Further backups should be fast as only changes are taken into account.

It is recommended to create a backup before any container update. By doing this, you will be safe regarding any possible complication during updates because you will be able to restore the whole instance with basically one click.

For local backups, the restore process should be pretty fast as rsync is used to restore the chosen backup which only transfers changed files and deletes additional ones. For remote borg backups, the whole backup archive is extracted from the remote, which depending on how clever `borg extract` is, may require downloading the whole archive.

If you connect an external drive to your host, and choose the backup directory to be on that drive, you are also kind of safe against drive failures of the drive where the docker volumes are stored on.

▶ How to do the above step for step

If you want to back up directly to a remote borg repository:

▶ How to do the above step for step

Backups can be created and restored in the AIO interface using the buttons `Create Backup` and `Restore selected backup`. Additionally, a backup check is provided that checks the integrity of your backups but it shouldn't be needed in most situations.

The backups themselves get encrypted with an encryption key that gets shown to you in the AIO interface. Please save that at a safe place as you will not be able to restore from backup without this key.

Daily backups can get enabled after the initial backup is done. Enabling this also allows to enable an option that allows to automatically update all containers, Nextcloud and its apps.

Be aware that this solution does not back up files and folders that are mounted into Nextcloud using the external storage app - but you can add further Docker volumes and host paths that you want to back up after the initial backup is done.

What is getting backed up by AIO's backup solution?

Backed up will get all important data of your Nextcloud AIO instance required to restore the instance, like the database, your files and configuration files of the mastercontainer and else. Files and folders that are mounted into Nextcloud using the external storage app are not getting backed up. There is currently no way to exclude the data directory because it would require hacks like running files:scan and would make the backup solution much more unreliable (since the database and your files/folders need to stay in sync). If you still don't want your datadirectory to be backed up, see [https://github.com/nextcloud/all-in-one#how-to-enable-automatic-updates-without-creating-a-backup-beforehand](#) for options (there is a hint what needs to be backed up in which order).

How to adjust borgs retention policy?

The built-in borg-based backup solution has by default a retention policy of `--keep-within=7d --keep-weekly=4 --keep-monthly=6`. See [https://borgbackup.readthedocs.io/en/stable/usage/prune.html](#) for what these values mean. You can adjust the retention policy by providing `--env BORG_RETENTION_POLICY="--keep-within=7d --keep-weekly=4 --keep-monthly=6"` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest`! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used) and customize the value to your fitting. ⚠️ Please make sure that this value is valid, otherwise backup pruning will bug out!

How to migrate from AIO to AIO?

If you have the borg backup feature enabled, you can copy it over to the new host and restore from the backup. This guide assumes the new installation data dir will be on `/mnt/datadir`, you can adjust the steps if it's elsewhere.

1. Set the DNS entry to 60 seconds TTL if applicable
2. On your current installation, use the AIO interface to:
    i. Update AIO and all containers
    ii. Stop all containers (from now on, your cloud is down)
    iii. Create a current borg backup

       iv. Note the path where the backups are stored and the encryption password

3. Navigate to the backup folder

4. Create archive of the backup so it's easier to copy: `tar -czvf borg.tar.gz borg`

5. Copy the archive over to the new host: `cp borg.tar.gz user@new.host:/mnt` . Make sure to replace `user` with your actual user and `new.host` with the IP or domain of the actual host. You can also use another way to copy the archive.

6. Switch to the new host

7. Go to the folder you put the backup archive and extract it with `tar -xf borg.tar.gz`

8. Follow the installation guide to create a new aio instance, but do not start the containers yet (the `docker run` or `docker compose up -d` command)

9. Change the DNS entry to the new host's IP

10. Configure your reverse proxy if you use one

11. Start the AIO container and open the new AIO interface in your browser

12. Make sure to save the newly generated passphrase and enter it in the next step

13. Select the "Restore former AIO instance from backup" option and enter the encryption password from the old backup and the path in which the extracted `borg` folder lies in (without the borg part) and hit `Submit location and password`

14. Choose the latest backup in the dropdown and hit `Restore selected backup`

15. Wait until the backup is restored

16. Start the containers in the AIO interface

Are remote borg backups supported?

Backing up directly to a remote borg repository is supported. This avoids having to store a local copy of your backups, supports append-only borg keys to counter ransomware and allows using the AIO interface to manage your backups.

Some alternatives, which do not have all the above benefits:

- Mount a network FS like SSHFS, SMB or NFS in the directory that you enter in AIO as backup directory
- Use rsync or rclone for syncing the borg backup archive that AIO creates locally to a remote target (make sure to lock the backup archive correctly before starting the sync; search for "aio-lockfile"; you can find a local example script here: https://github.com/nextcloud/all-in-one#sync-the-backup-regularly-to-another-drive)
- You can find a well written guide that uses rclone and e.g. BorgBase for remote backups here: #2247
- Here is another one that utilizes borgmatic and BorgBase for remote backups: #4391
- create your own backup solution using a script and borg, borgmatic or any other to backup tool for backing up to a remote target (make sure to stop and start the AIO containers correctly following https://github.com/nextcloud/all-in-one#how-to-enable-automatic-updates-without-creating-a-backup-beforehand)

Failure of the backup container in LXC containers

If you are running AIO in a LXC container, you need to make sure that FUSE is enabled in the LXC container settings. Also, if using Alpine Linux as host OS, make sure to add fuse via `apk add fuse` . Otherwise the backup container will not be able to start as FUSE is required for it to work.

How to create the backup volume on Windows?

As stated in the AIO interface, it is possible to use a docker volume as backup target. Before you can use that, you need to create it first. Here is an example how to create one on Windows:

```
docker volume create ^
--driver local ^
--name nextcloud_aio_backupdir ^
-o device="/host_mnt/e/your/backup/path" ^
-o type="none" ^
-o o="bind"
```

In this example, it would mount `E:\your\backup\path` into the volume so for a different location you need to adjust `/host_mnt/e/your/backup/path` accordingly. Afterwards enter `nextcloud_aio_backupdir` in the AIO interface as backup location.

Pro-tip: Backup archives access

You can open the BorgBackup archives on your host by following these steps:
(instructions for Ubuntu Desktop)

Alternatively, there is now a community container that allows to access your backups in a web session: https://github.com/nextcloud/all-in-one/tree/main/community-containers/borgbackup-viewer.

```
# Install borgbackup on the host
sudo apt update && sudo apt install borgbackup

# In any shell where you use borg, you must first export this variable
# If you are using the default backup location /mnt/backup/borg
export BORG_REPO='/mnt/backup/borg'
# or if you are using a remote repository
export BORG_REPO='user@host:/path/to/repo'

# Mount the archives to /tmp/borg
sudo mkdir -p /tmp/borg && sudo borg mount "$BORG_REPO" /tmp/borg

# After entering your repository key successfully, you should be able to access all archives in /tmp/borg
# You can now do whatever you want by syncing them to a different place using rsync or doing other things
# E.g. you can open the file manager on that location by running:
xhost +si:localuser:root && sudo nautilus /tmp/borg

# When you are done, simply close the file manager and run the following command to unmount the backup archives:
sudo umount /tmp/borg
```

Delete backup archives manually

You can delete BorgBackup archives on your host manually by following these steps:
(instructions for Debian based OS' like Ubuntu)

Alternatively, there is now a community container that allows to access your backups in a web session: https://github.com/nextcloud/all-in-one/tree/main/community-containers/borgbackup-viewer.

```
# Install borgbackup on the host
sudo apt update && sudo apt install borgbackup

# In any shell where you use borg, you must first export this variable
# If you are using the default backup location /mnt/backup/borg
export BORG_REPO='/mnt/backup/borg'
# or if you are using a remote repository
export BORG_REPO='user@host:/path/to/repo'

# List all archives (if you are using the default backup location /mnt/backup/borg)
sudo borg list

# After entering your repository key successfully, you should now see a list of all backup archives
# An example backup archive might be called 20220223_174237-nextcloud-aio
# Then you can simply delete the archive with:
sudo borg delete --stats --progress "::20220223_174237-nextcloud-aio"

# If borg 1.2.0 or higher is installed, you then need to run borg compact in order to clean up the freed space
sudo borg --version
# If version number of the command above is higher than 1.2.0 you need to run the command below:
sudo borg compact
```

After doing so, make sure to update the backup archives list in the AIO interface!
You can do so by clicking on the `Check backup integrity` button or `Create backup` button.

Sync local backups regularly to another drive

For increased backup security, you might consider syncing the local backup repository regularly to another drive.

To do that, first add the drive to `/etc/fstab` so that it is able to get automatically mounted and then create a script that does all the things automatically. Here is an example for such a script:

▶ Click here to expand

You can simply copy and paste the script into a file e.g. named `backup-script.sh` e.g. here: `/root/backup-script.sh`. Do not forget to modify the variables to your requirements!

Afterwards apply the correct permissions with `sudo chown root:root /root/backup-script.sh` and `sudo chmod 700 /root/backup-script.sh`. Then you can create a cronjob that runs e.g. at `20:00` each week on Sundays like this:

1. Open the cronjob with `sudo crontab -u root -e` (and choose your editor of choice if not already done. I'd recommend nano).
2. Add the following new line to the crontab if not already present: `0 20 * * 7 /root/backup-script.sh` which will run the script at 20:00 on Sundays each week.
3. save and close the crontab (when using nano are the shortcuts for this `Ctrl + o` -> `Enter` and close the editor with `Ctrl + x`).

How to exclude Nextcloud's data directory or the preview folder from backup?

In order to speed up the backups and to keep the backup archives small, you might want to exclude Nextcloud's data directory or its preview folder from backup.

> ⚠ Warning
>
> However please note that you will run into problems if the database and the data directory or preview folder get out of sync. **So please only read further, if you have an additional external backup of the data directory!** See this guide for example.

> 💡 Tip
>
> A better option is to use the external storage app inside Nextcloud as the data connected via the external storage app is not backed up by AIO's backup solution. See this documentation on how to configure the app.

If you still want to proceed, you can exclude the data directory by simply creating a `.noaiobackup` file in the root directory of the specified `NEXTCLOUD_DATADIR` target. The same logic is implemented for the preview folder that is located inside the data directory, inside the `appdata_*/preview` folder. So simply create a `.noaiobackup` file in there if you want to exclude the preview folder.

After doing a restore via the AIO interface, you might run into problems due to the data directory and database being out of sync. You might be able to fix this by running `occ files:scan --all` and `occ maintenance:repair` and `occ files:scan-app-data`. See https://github.com/nextcloud/all-in-one#how-to-run-occ-commands. If only the preview folder is excluded, the command `occ files:scan-app-data preview` should be used.

How to stop/start/update containers or trigger the daily backup from a script externally?

> ⚠ Warning
>
> The below script will only work after the initial setup of AIO. So you will always need to first visit the AIO interface, type in your domain and start the containers the first time or restore an older AIO instance from its borg backup before you can use the script.

You can do so by running the `/daily-backup.sh` script that is stored in the mastercontainer. It accepts the following environment variables:

- `AUTOMATIC_UPDATES` if set to `1`, it will automatically stop the containers, update them and start them including the mastercontainer. If the mastercontainer gets updated, this script's execution will stop as soon as the mastercontainer gets stopped. You can then wait until it is started again and run the script with this flag again in order to update all containers correctly afterwards.
- `DAILY_BACKUP` if set to `1`, it will automatically stop the containers and create a backup. If you want to start them again afterwards, you may have a look at the `START_CONTAINERS` option.
- `START_CONTAINERS` if set to `1`, it will automatically start the containers without updating them.
- `STOP_CONTAINERS` if set to `1`, it will automatically stop the containers.
- `CHECK_BACKUP` if set to `1`, it will start the backup check. This is not allowed to be enabled at the same time like `DAILY_BACKUP`. Please be aware that this option is non-blocking which means that the backup check is not done when the process is finished since it only start the borgbackup container with the correct configuration.

One example for this would be `sudo docker exec -it --env DAILY_BACKUP=1 nextcloud-aio-mastercontainer /daily-backup.sh` , which you can run via a cronjob or put it in a script.

> ⓘ **Note**
>
> None of the option returns error codes. So you need to check for the correct result yourself.

How to disable the backup section?

If you already have a backup solution in place, you may want to hide the backup section. You can do so by adding `--env AIO_DISABLE_BACKUP_SECTION=true` to the docker run command of the mastercontainer (but before the last line `ghcr.io/nextcloud-releases/all-in-one:latest` ! If it was started already, you will need to stop the mastercontainer, remove it (no data will be lost) and recreate it using the docker run command that you initially used).

Addons

Fail2ban

You can configure your server to block certain ip-addresses using fail2ban as bruteforce protection. Here is how to set it up: https://docs.nextcloud.com/server/stable/admin_manual/installation/harden_server.html#setup-fail2ban. The logpath of AIO is by default `/var/lib/docker/volumes/nextcloud_aio_nextcloud/_data/data/nextcloud.log` . Do not forget to add `chain=DOCKER-USER` to your nextcloud jail config (`nextcloud.local`) otherwise the nextcloud service running on docker will still be accessible even if the IP is banned. Also, you may change the blocked ports to cover all AIO ports: by default `80,443,8080,8443,3478` (see this). Apart from that there is now a community container that can be added to the AIO stack: https://github.com/nextcloud/all-in-one/tree/main/community-containers/fail2ban

LDAP

It is possible to connect to an existing LDAP server. You need to make sure that the LDAP server is reachable from the Nextcloud container. Then you can enable the LDAP app and configure LDAP in Nextcloud manually. If you don't have a LDAP server yet, recommended is to use this docker container: https://hub.docker.com/r/nitnelave/lldap. Make sure here as well that Nextcloud can talk to the LDAP server. The easiest way is by adding the LDAP docker container to the docker network `nextcloud-aio` . Then you can connect to the LDAP container by its name from the Nextcloud container. There is now a community container which allows to easily add LLDAP to AIO: https://github.com/nextcloud/all-in-one/tree/main/community-containers/lldap

Netdata

Netdata allows you to monitor your server using a GUI. You can install it by following https://learn.netdata.cloud/docs/agent/packaging/docker#create-a-new-netdata-agent-container. Apart from that there is now a way for the community to add containers: #392 (comment)

USER_SQL

If you want to use the user_sql app, the easiest way is to create an additional database container and add it to the docker network `nextcloud-aio` . Then the Nextcloud container should be able to talk to the database container using its name.

phpMyAdmin, Adminer or pgAdmin

It is possible to install any of these to get a GUI for your AIO database. The pgAdmin container is recommended. You can get some docs on it here: https://www.pgadmin.org/docs/pgadmin4/latest/container_deployment.html. For the container to connect to the aio-database, you need to connect the container to the docker network `nextcloud-aio` and use `nextcloud-aio-database` as database host, `oc_nextcloud` as database username and the password that you get when running `sudo docker exec nextcloud-aio-nextcloud grep dbpassword config/config.php` as the password. Apart from that there is now a way for the community to add containers: #3061 (comment)

Mail server

You can configure one yourself by using either of these four recommended projects: Docker Mailserver, Mailu, Maddy Mail Server, Mailcow or Stalwart. There is now a community container which allows to easily add Stalwart Mail server to AIO: https://github.com/nextcloud/all-in-one/tree/main/community-containers/stalwart

Miscellaneous

Requirements for integrating new containers

For integrating new containers, they must pass specific requirements for being considered to get integrated in AIO itself. Even if not considered, we may add some documentation on it. Also there is this now: https://github.com/nextcloud/all-in-one/tree/main/community-containers#community-containers

What are the requirements?

1. New containers must be related to Nextcloud. Related means that there must be a feature in Nextcloud that gets added by adding this container.
2. It must be optionally installable. Disabling and enabling the container from the AIO interface must work and must not produce any unexpected side-effects.
3. The feature that gets added into Nextcloud by adding the container must be maintained by the Nextcloud GmbH.
4. It must be possible to run the container without big quirks inside docker containers. Big quirks means e.g. needing to change the capabilities or security options.
5. The container should not mount directories from the host into the container: only docker volumes should be used.
6. The container must be usable by more than 90% of the users (e.g. not too high system requirements and such)
7. No additional setup should be needed after adding the container - it should work completely out of the box.
8. If the container requires being exposed, only subfolders are supported. So the container should not require its own (sub-)domain and must be able to run in a subfolder.

Update policy

This project values stability over new features. That means that when a new major Nextcloud update gets introduced, we will wait at least until the first patch release, e.g. `24.0.1` is out before upgrading to it. Also we will wait with the upgrade until all important apps are compatible with the new major version. Minor or patch releases for Nextcloud and all dependencies as well as all containers will be updated to new versions as soon as possible but we try to give all updates first a good test round before pushing them. That means that it can take around 2 weeks before new updates reach the `latest` channel. If you want to help testing, you can switch to the `beta` channel by following this documentation which will also give you the updates earlier.

How often are update notifications sent?

Releases 204

🏷️ **v10.11.0** (Latest)
2 weeks ago

+ 203 releases

Contributors 142

+ 128 contributors

Languages

● **PHP** 51.8% ● **Shell** 26.0% ● **Twig** 12.1% ● **Dockerfile** 4.9% ● **JavaScript** 3.0% ● **CSS** 2.2%