Docker: Nextcloud AIO

Nextcloud is a fantastic Software for storing and sharing Files plus PIM- Data. With Nextcloud All In One (AIO) you can even have your own private Hosting of Videocalls plus dozens of Features for Teaming without any other Company beiing involved.

As you do have Docker running as described before, you can easyly install a running Instance of Nextcloud to you PC. Heres the original doc:
https://github.com/nextcloud/all-in-one

Docker Network

To have IPV6 support, Docker needs a local IPV6 Networkrange to use.

So you need to edit the Dockers ~/.config/docker/daemon.json to be like this:

Hint: If that file is missing, just create it.

```
{
 "ipv6": true,
 "fixed-cidr-v6": "fdff:6785:1::/48",
 "iptables": true,
 "ip6tables": true,
 "log-opts": {
  "max-size": "10m",
  "max-file": "5"
 }
}
```

Network preparations

As i do have IPV6 only to reach the server, it is very important to set up networking the right way to make nextcloud-aio work.

Per default, only ipv4 networking will be set up by nextcloud-aio. So the setup will check the adress and will fail.

So you need to setup a network first, that is called exactly „nextcloud-aio“ with ipv6.

To Set up the network in Docker, here for example use the script `docker-aio-network.sh`

```
#!/bin/bash
#recreate network for docker with ipv6
docker network remove nextcloud-aio
#Use the fixed-cidr-v6 from dockers daemon.json for the subnet and
#Adress :1 for the gateway
docker network create --subnet="fdff:6785:2::/48" --gateway="fdff:6785:2::1" --driver bridge --ipv6 nextcloud-aio
```

The subnet and gateway in daemon.json are free to choose, but make sure it does not interfere with some existing network. If you are not sure, use the given ones, as they are locally and should be free for docker.

Firewall

According to this: https://github.com/nextcloud/all-in-one?tab=readme-ov-file#how-to-resolve-firewall-problems-with-fedora-linux-rhel-os-centos-suse-linux-and-others there are problems with firewalld and docker which leads to Docker Containers not beeing able to communicate to each other.

So, do this as the Docker- User (in rootless- mode) before:

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

Docker Network driver

There may be other problems with networking, so use another driver for networking like described here: https://github.com/nextcloud/all-in-one/issues/4621

In general, the default slirp4netns- driver is: a. slow b. not able to uses ipv6 c. not able to do port forwarding, which maybe essential

Pasta driver

A quite new and high performance networking driver with good functionality is pasta.

Pasta needs to be installed locally on the host `zypper install pasta`

After that, you need to change the systemd config für docker:

```
docker@pcserver2023:~> systemctl --user edit docker
```

and edit the file like this:

```
[Service]
Environment="DOCKERD_ROOTLESS_ROOTLESSKIT_NET=pasta"
Environment="DOCKERD_ROOTLESS_ROOTLESSKIT_PORT_DRIVER=implicit"
Environment="DOCKERD_ROOTLESS_ROOTLESSKIT_FLAGS=--ipv6"
```

now do

```
docker@pcserver2023:~> systemctl daemon-reload

docker@pcserver2023:~> systemctl --user stop docker

docker@pcserver2023:~> systemctl --user start docker
```

You shold now have a fast network driver with port forwarding.

Dockers Yaml

In your Docker- Compose- Drirectory, create a new Directory called `nextcloud_aoi` and put the `docker-compose.yml` in it. You can finde some good documented example here: https://github.com/nextcloud/all-in-one/blob/main/compose.yaml

Adjustements:

- For long Fileuploads, set NEXTCLOUD_MAX_TIME=3600 = one Hour to a higher value, like 14400 = four hours
- Also maybe NEXTCLOUD_UPLOAD_LIMIT=12G and
- maybe NEXTCLOUD_MEMORY_LIMIT=2048M

AIO Webinterface

It is important to understand, that the Docker- Service itself is NOT a working Nextcloud- Instance!

The Service of this YML - called `nextcloud-aio-mastercontainer` is only a WEB- Interface to setup and maintain the Nextcloud- Services. Therefore it uses the Docker internal API, which is accessed by the Docker- Socket internally.

So what you gain first, is a new Webserver on a seperate Port (default: 8080), wich will be the Administration Endpoint for you.

To access that Server from another PC, i STRONGLY DO NOT ADVISE to open the Firewall- Port of your Host and to make that service available to the internet!

Instead, you should use SSH Port Forwarding to administrate the Service. To have that, you maybe use ssh like this:

```
ssh -L 8080:IPofyoursshserver:8080 docker@NAMEofyoursshserver
```

After that, you can access the AIO- Mastercontainer WEB- GUI on https://localhost:8080

Caddy Service

Now you need to add the Nextcloud- Service to you Caddyfile (https://obel1x.de/dokuwiki/doku.php?id=content:serverbasics:docker-caddy#caddy_configuration)

Add those lines First:

```
https://nextcloud.domain.tld:443 {
       header Strict-Transport-Security max-age=31536000;
#Large fileuploads
       request_body {
           max_size 10240M
```

```
        }
#       reverse_proxy nextcloud-aio-apache:11000
#For install Domaincheck needed:
        reverse_proxy nextcloud-aio-domaincheck:11000
}
```

Mind, that when starting Nextcloud-AIO, the service nextcloud-aio-domaincheck will be setup. After installation has succeeded, the service will be shutdown and nextcloud-aio-apache will take over. On a Host, having rootful docker, this will work with localhost:11000 as destination, but not in rootless docker.

So we need this small quirk to work around it.

After you have setup the Configuration in Nextcloud, when installing and starting the Services, comment out that line and uncomment the apache- line. Than restart Caddy - you won't need that line again.

Final Nextcloud- Setup

Now, ssh on your Host and go to https://localhost:8080

and finish everything on that WEB- GUI. Mind, that installing the Services will take a long time. Don't panik if nothing seems to happen. Just let it do one hour or more.

Enjoy your fully flagged Nextcloud.

IMPORTANT : You should install the Container https://github.com/nextcloud/all-in-one/tree/main/community-containers/borgbackup-viewer

And read about Backups and restore VERY CAREFULLY here: docker-backup

The next step would now be to integrate Nextcloud with your IPA- Domain.

Enable SSO to Logon with FreeIPA/Authentik

Before proceeding, you should have integrated you Client to the domain as written in docker-freeipa , you should be logged into your PC as freeipa- user.

Than, check, that Authentik- Kerberos is working as described at docker-authentik and that you can authenticate in Authentik only by using the Kerberos- Symbol at Authentik- Logon- Page.

If this is working, check this docs to integrate Nextcloud with Authentik/SSO: https://docs.goauthentik.io/integrations/services/nextcloud/

Attention: to have the UserID of FreeIPA be used by Nextcloud, i had to:

- In Authentik / Provider Nextcloud: under Advance Protocol Settings: Subject mode = Based on user's username
- And in ODIC- Settings of Nextcloud, use „sub“ as userid-mapping.
- Turn off „Use unique Userids“ and Turn on Group Provisioniong so that your Admins will become NC- Admins

All other Settungs either made Admins not beeing Admins in NC or had hashed- userids in Nexctcloud, which make it complicated to get acces to NC with your PCs UserID.

Disable User Logon-Screen

After you can logon with authentik, disable the user-input screen by opening container shell of `nextcloud-aio-nextcloud` container and enter:

```
containerid:/var/www/html# sudo -u www-data php /var/www/html/occ config:app:set --value=0 user_oidc
allow_multiple_user_backends
```

Remember, that you now need to go to

`http://nextcloud.domain.tld/login?direct=1`

to logon to nextcloud with the local admin.