

Docker

- [Updating container using docker-compose](#)
- [Setting up Apache Guacamole using Docker](#)
- [Troubleshooting Docker](#)
- [Paperless: ProtonMail Bridge](#)
- [Your URLs](#)
- [Cloudflare](#)
- [Updating a docker container running on QNAP using SSH](#)

Updating container using docker-compose

- docker compose pull
- docker compose up -d --remove-orphans
- docker image prune

To list all containers:

- `docker ps -a`

Setting up Apache Guacamole using Docker

Visit: <https://github.com/boschkundendienst/guacamole-docker-compose>

Follow instructions which in short are:

1. git clone "https://github.com/boschkundendienst/guacamole-docker-compose.git"
2. cd guacamole-docker-compose
3. Edit docker-compose.yml to suite your environment
4. ./prepare.sh
5. docker compose up -d

If you want to add TOTP then do the following:

1. In your docker-compose.yml file add
 - Environment variable: GUACAMOLE_HOME=/config/guacamole
 - Mount ./guacamole:/config/guacamole
2. Create in your docker directory
 - create the following folders guacamole/extensions/
 - wget <https://downloads.apache.org/guacamole/1.6.0/binary/guacamole-auth-totp-1.6.0.tar.gz> (or matching version to your guacamole install) and extract the guacamole-auth-totp-1.6.0.jar to the above extensions folder

After adding the above extension, restart the container.

Troubleshooting Docker

Check Environment Variables Inside the Container:

```
sudo docker exec -it container-name env
```

Review Logs:

```
sudo docker logs container-name
```

```
sudo docker-compose up -d && sudo docker-compose logs -f
```

Paperless: ProtonMail Bridge

Source code: [shenxn/protonmail-bridge-docker: ProtonMail IMAP/SMTP Bridge Docker container \(github.com\)](https://github.com/shenxn/protonmail-bridge-docker)

Change to the Paperless docker directory

```
sudo docker compose down
sudo docker run --rm -it -v protonmail:/root shenxn/protonmail-bridge init
```

Enter "login" to connect the ProtonMail account. Once complete, type "info" to get the login details that will be used in the Paperless mail interface.

Exit the instance (using control-c if necessary).

Bring paperless up again.

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

Configure and test the mail settings

Edit mail account ID: 2

Name	<input type="text" value="shenxn@protonmail.com"/>	Username	<input type="text" value="shenxn@protonmail.com"/>
IMAP Server	<input type="text" value="protonmail-bridge"/>	Password	<input type="password" value=""/>
IMAP Port	<input type="text" value="143"/>	<input type="checkbox"/> Password is token <small>Check if the password above is a token used for authentication</small>	
IMAP Security	<input type="text" value="No encryption"/>	Character Set	<input type="text" value="UTF-8"/>

Your URLs

```
version: '3.1'
```

```
services:
```

```
  yourls:
```

```
    image: yourls
```

```
    restart: always
```

```
    ports:
```

```
      - 8080:80
```

```
    volumes:
```

```
      - /home/sysadmin/yourls:/var/www/html
```

```
    environment:
```

```
      YOURLS_DB_PASS: dbpwd
```

```
      YOURLS_SITE: https://192.168.20.40:8080/
```

```
      YOURLS_USER: osgadmin
```

```
      YOURLS_PASS: yourlspwd
```

```
  mysql:
```

```
    image: mysql
```

```
    restart: always
```

```
    environment:
```

```
      MYSQL_ROOT_PASSWORD: dbpwd
```

```
      MYSQL_DATABASE: yourls
```

Cloudflare

```
version: "3.9"
services:

  tunnel:
    container_name: cloudflared-tunnel
    image: cloudflare/cloudflared
    restart: unless-stopped
    command: tunnel run
    environment:
      - TUNNEL_TOKEN=token.....
```

Updating a docker container running on QNAP using SSH

SSH enabled on your QNAP NAS

1. Identify the Container and Image Name

Option A: Using Container Station GUI

1. Open **Container Station**.
2. Navigate to the **Containers** tab.
3. Click on the target container to view details.
4. Locate the **Image** field — this is the image used (e.g., `linuxserver/nextcloud:latest`).

Option B: Using SSH

```
ssh admin@<your-nas-ip>  
docker ps
```

Copy the container name (e.g., `nextcloud`) and run:

```
docker inspect --format='{{.Config.Image}}' <container-name>
```

This returns the image name, e.g.:

```
linuxserver/nextcloud:latest
```

2. Back Up Container Volumes (Optional but Recommended)

Check volume mappings:

```
docker inspect <container-name> | grep -A 10 "Mounts"
```

Back up the volume path if necessary. For example, if a volume is mounted to

```
/share/Container/nextcloud/config
```

, back that up using QNAP File Station or `rsync`.

3. Pull the Latest Docker Image

```
docker pull <image-name>
```

Example:

```
docker pull linuxserver/nextcloud:latest
```

4. Stop and Remove the Old Container

```
docker stop <container-name>
```

```
docker rm <container-name>
```

Example:

```
docker stop nextcloud
```

```
docker rm nextcloud
```

“ ⚠ This does **not** delete the image or volume data.

5. Recreate the Container with Same Settings

Get Existing Settings (Ports, Volumes, Env Vars)

Use:

```
docker inspect <container-name>
```

Note the following:

- Port mappings
 - Volume mounts
 - Environment variables
-

Re-run the Container

Example:

```
docker run -d \  
  --name nextcloud \  
  -e PUID=1000 \  
  -e PGID=1000 \  
  -e TZ=Europe/London \  
  -p 8080:80 \  
  -v /share/Container/nextcloud/config:/config \  
  -v /share/Container/nextcloud/data:/data \  
  linuxserver/nextcloud:latest
```

“ Replace volume paths, ports, and environment variables based on what you had before.

6. Verify Everything Works

- Use:
`docker ps`
to confirm the container is running.
- Check logs:
`docker logs -f <container-name>`
- Access the app via web browser or API to confirm it's working.

7. (Optional) Remove Old Images

List unused images:

```
docker images
```

Clean up dangling images:

```
docker image prune
```

Or remove a specific old image manually:

```
docker rmi <image-id>
```

Bonus: Automatically Extract and Re-run a Container

To automatically generate a `docker run` command:

```
docker run --rm \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  red5d/docker-autocompose <container-name> > recreate-container.yml
```

Then review or convert the output back into a run command.