# 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.

---

Revision #1
Created 2025-05-28 20:17:52 UTC by Admin
Updated 2025-05-28 20:24:46 UTC by Admin