

Back up Docker container data folders (stop, archive, restart)

For apps whose data can't be copied safely while they are running, the simple, reliable method is: stop the container, tar its data folder, then start it again. Do this one container at a time, so each app is only down for a moment.

Assumptions

Each container's data lives in `<base>/<container-name>/` (for example `/srv/docker/sonarr`), and backups go to a mounted share at `/mnt/backup`.

The script

```
#!/bin/bash
set -uo pipefail                # no -e: we must always restart the container
BASE=/srv/docker
DEST=/mnt/backup/docker/$(hostname -s)/$(date +%F)
KEEP=14
mountpoint -q /mnt/backup || { echo "backup share not mounted"; exit 1; }
mkdir -p "$DEST"
for c in $(docker ps --format '{{.Names}}'); do
    [[ -d $BASE/$c ]] || { echo "skip $c (no data folder)"; continue; }
    docker stop -t 120 "$c" >/dev/null || { echo "could not stop $c"; continue; }
    if tar -czf "$DEST/$c.tar.gz.part" -C "$BASE" "$c"; then
        mv "$DEST/$c.tar.gz.part" "$DEST/$c.tar.gz"
    else
        echo "backup of $c failed"; rm -f "$DEST/$c.tar.gz.part"
    fi
done
```

```
fi
docker start "$c" >/dev/null || echo "WARNING: $c did not restart"
done
find "${DEST%/*}" -type f -name '*.tar.gz' -mtime +$KEEP -delete
find "${DEST%/*}" -mindepth 1 -type d -empty -delete
```

Why no `set -e`

With `set -e`, a failing `tar` (or `tar`'s exit code 1 for "file changed while reading") ends the script **before** `docker start`, leaving the app down. Handle errors explicitly, or add a `trap` that restarts the current container.

Skip containers whose folder isn't there

Check that the data folder exists **before** stopping the container. Otherwise a wrong `BASE` path means every container is stopped and started for nothing.

Stacks with databases

If an app and its database are separate containers, stop the app first, then the database, back up both, and start them in reverse order. Or use a proper database dump; see *Back up Docker app SQLite databases without stopping containers*.

Schedule

```
0 3 * * * root /usr/local/bin/docker_backup.sh >> /var/log/docker-backup.log 2>&1
```

Restore

`docker stop app`, then `tar -xzf app.tar.gz -C /srv/docker`, then `docker start app`.

Revision #1

Created 24 October 2026 13:29:32 by Admin

Updated 24 October 2026 13:29:32 by Admin