

Lab 3 - Volumes

Use volumes

Tips

- Use `docker container inspect ...`

Existing volumes

1. run `docker container run --name couchdb -d -p 5984:5984 couchdb:2.1`

Unable to find image 'couchdb:latest' locally

latest: Pulling from library/couchdb

8b969e402096: Pull complete

ec13befd006f: Pull complete

79d2aef50f2c: Pull complete

2e5301367e88: Pull complete

4bb50d4b8156: Pull complete

e2f67cda63d6: Pull complete

6cf41befbfe7: Pull complete

72b50ca53611: Pull complete

8d1d3615bc1b: Pull complete

923cef98fe32: Pull complete

Digest: sha256:432e99b1dbb8bd996b2af964b4731633956e9635548e82b037bda94bb1835193

Status: Downloaded newer image for couchdb:latest

91cbacec405af6ba4de2b808961defb70b40cf650b53c76f61ed3e783be1958b

2. Check existing volumes

- `docker volume ls`

local f11bd9b5df487756522e4519b460a58934cebee0699a03a026ce30ee7c731a93

local lamafama_n8n_data

local lamafama_postgres_data

1. Why there is already a volume ?

CouchDB (like many databases) automatically creates a volume for its data persistence (so data isn't lost when container stops).

3. Identify the volume that is used by `couchdb`

- `docker inspect couchdb`

```
"Mounts": [  
  {
```

```

        "Type": "volume",
        "Name":
        "f11bd9b5df487756522e4519b460a58934cebee0699a03a026ce30ee7c731a93",
        "Source":
        "/var/lib/docker/volumes/f11bd9b5df487756522e4519b460a58934cebee069
        9a03a026ce30ee7c731a93/_data",
        "Destination": "/opt/couchdb/data",
        "Driver": "local",
        "Mode": "",
        "RW": true,
        "Propagation": ""
    }

```

4. Mount the identified volume to busybox

- `docker run -it --rm -v f11bd9b5df487756522e4519b460a58934cebee0699a03a026ce30ee7c731a93:/opt/couchdb/data busybox`

Unable to find image 'busybox:latest' locally

latest: Pulling from library/busybox

499bcf3c8ead: Pull complete

Digest: sha256:ab33eacc8251e3807b85bb6dba570e4698c3998eca6f0fc2ccb60575a563ea74

Status: Downloaded newer image for busybox:latest

/ #

/ #

/ #

/ #

5. Check files inside `/opt/couchdb/data`

- `ls /opt/couchdb/data`
it is empty

6. Stop couchdb

- `docker stop couchdb`

7. Delete the volume

- `docker container rm couchdb`
- `docker volume rm f11bd9b5df487756522e4519b460a58934cebee0699a03a026ce30ee7c731a93`

8. Check that the volume has been deleted

- `docker volume ls`

DRIVER VOLUME NAME

local lamafama_n8n_data

Create a named volume

1. Create a volume named : `couchdb_vol`

- `docker volume create couchdb_vol`

2. Run `couchdb` with the created volume

- `docker run -d --name couchdb2 -p 5985:5984 -v couchdb_vol:/opt/couchdb/data couchdb:latest`

`vol:/opt/couchdb/data couchdb:latest`

`074265e15c65cf99569076c355ab014ab22fde50213981bfda81296769cd144f`

3. Inspect the container and look at `Mounts` that `couchdb_vol` is used

- `docker inspect couchdb2`

```
"Mounts": [  
{  
  "Type": "volume",  
  "Name": "couchdb_vol",  
  "Source": "/var/lib/docker/volumes/couchdb_vol/_data",  
  "Destination": "/opt/couchdb/data",  
  "Driver": "local",  
  "Mode": "z",  
  "RW": true,  
  "Propagation": ""  
}]
```

Mount directory

1. Mount `/var/lib/docker` from host to `/dck` into a busybox container

- `docker run -it --rm -v /var/lib/docker:/dck busybox`

2. Check at `ls /dck/volumes/couchdb_vol/_data` inside the container to verify that `couchdb_vol` is available

```
~ # cd /dck/volumes/couchdb_vol/_data  
/dck/volumes/couchdb_vol/_data # ls  
/dck/volumes/couchdb_vol/_data # ls -a  
...
```

SideCar Pattern

1. Create a directory named `sidecar` with `mkdir`

- `mkdir sidecar`

2. Run Busybox

1. Command: `sh -c 'while true; do date >> /dck/date.log; sleep 1; done'`
2. Volume to mount: `$(pwd)/sidecar:/dck`
3. Name: `gen_date`
4. State: detached

- `docker run -d --name gen_date -v "$(pwd)/sidecar:/dck" busybox sh -c "while true; do date >> /dck/date.log; sleep 1; done"`

- `docker ps`

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

188f3b3cb40c busybox "sh -c 'while true; ..." 2 minutes ago Up 2 minutes gen_date

a05f6922344f my_awesome_image "/docker-entrypoint...." 17 hours ago Up 17 hours

0.0.0.0:8080->80/tcp, [::]:8080->80/tcp mynginx2

7b5969362aa8 nginx "/docker-entrypoint...." 17 hou

3. Check the content of `sidecar/date.log` with `cat`

- `cat sidecar/date.log`

Thu Aug 28 08:27:43 UTC 2025

Thu Aug 28 08:27:44 UTC 2025

Thu Aug 28 08:27:45 UTC 2025

Thu Aug 28 08:27:46 UTC 2025

Thu Aug 28 08:27:47 UTC 2025

Thu Aug 28 08:27:48 UTC 2025

Thu Aug 28 08:27:49 UTC 2025

Thu Aug 28 08:27:50 UTC 2025

Thu Aug 28 08:27:51 UTC 2025

Thu Aug 28 08:27:52 UTC 2025

Thu Aug 28 08:27:53 UTC 2025

Thu Aug 28 08:27:54 UTC 2025

Thu Aug 28 08:27:55 UTC 2025

Thu Aug 28 08:27:56 UTC 2025

Thu Aug 28 08:27:57 UTC 2025

Thu Aug 28 08:27:58 UTC 2025

Thu Aug 28 08:27:59 UTC 2025

Thu Aug 28 08:28:00 UTC 2025

Thu Aug 28 08:28:01 UTC 2025

Thu Aug 28 08:28:02 UTC 2025

Thu Aug 28 08:28:03 UTC 2025

Thu Aug 28 08:28:04 UTC 2025

Thu Aug 28 08:28:05 UTC 2025

Thu Aug 28 08:28:06 UTC 2025

Thu Aug 28 08:28:07 UTC 2025

Thu Aug 28 08:28:08 UTC 2025

Thu Aug 28 08:28:09 UTC 2025

Thu Aug 28 08:28:10 UTC 2025
Thu Aug 28 08:28:11 UTC 2025
Thu Aug 28 08:28:12 UTC 2025
Thu Aug 28 08:28:13 UTC 2025
Thu Aug 28 08:28:14 UTC 2025
Thu Aug 28 08:28:15 UTC 2025
Thu Aug 28 08:28:16 UTC 2025
Thu Aug 28 08:28:17 UTC 2025
Thu Aug 28 08:28:18 UTC 2025
Thu Aug 28 08:28:19 UTC 2025
Thu Aug 28 08:28:20 UTC 2025
Thu Aug 28 08:28:21 UTC 2025
Thu Aug 28 08:28:22 UTC 2025
Thu Aug 28 08:28:23 UTC 2025
Thu Aug 28 08:28:24 UTC 2025
Thu Aug 28 08:28:25 UTC 2025
Thu Aug 28 08:28:26 UTC 2025
Thu Aug 28 08:28:27 UTC 2025
Thu Aug 28 08:28:28 UTC 2025
Thu Aug 28 08:28:29 UTC 2025
Thu Aug 28 08:28:30 UTC 2025
Thu Aug 28 08:28:31 UTC 2025
Thu Aug 28 08:28:32 UTC 2025
Thu Aug 28 08:28:33 UTC 2025
Thu Aug 28 08:28:34 UTC 2025
Thu Aug 28 08:28:35 UTC 2025
Thu Aug 28 08:28:36 UTC 2025
Thu Aug 28 08:28:37 UTC 2025
Thu Aug 28 08:28:38 UTC 2025
Thu Aug 28 08:28:39 UTC 2025
Thu Aug 28 08:28:40 UTC 2025
Thu Aug 28 08:28:41 UTC 2025
Thu Aug 28 08:28:42 UTC 2025
Thu Aug 28 08:28:43 UTC 2025
Thu Aug 28 08:28:44 UTC 2025
Thu Aug 28 08:28:45 UTC 2025
Thu Aug 28 08:28:47 UTC 2025
Thu Aug 28 08:28:48 UTC 2025
Thu Aug 28 08:28:49 UTC 2025
Thu Aug 28 08:28:50 UTC 2025
Thu Aug 28 08:28:51 UTC 2025
Thu Aug 28 08:28:52 UTC 2025
Thu Aug 28 08:28:53 UTC 2025
Thu Aug 28 08:28:54 UTC 2025
Thu Aug 28 08:28:55 UTC 2025

Thu Aug 28 08:28:56 UTC 2025
Thu Aug 28 08:28:57 UTC 2025
Thu Aug 28 08:28:58 UTC 2025
Thu Aug 28 08:28:59 UTC 2025
Thu Aug 28 08:29:00 UTC 2025
Thu Aug 28 08:29:01 UTC 2025
Thu Aug 28 08:29:02 UTC 2025
Thu Aug 28 08:29:03 UTC 2025
Thu Aug 28 08:29:04 UTC 2025
Thu Aug 28 08:29:05 UTC 2025
Thu Aug 28 08:29:06 UTC 2025
Thu Aug 28 08:29:07 UTC 2025
Thu Aug 28 08:29:08 UTC 2025
Thu Aug 28 08:29:09 UTC 2025
Thu Aug 28 08:29:10 UTC 2025
Thu Aug 28 08:29:11 UTC 2025
Thu Aug 28 08:29:12 UTC 2025
Thu Aug 28 08:29:13 UTC 2025
Thu Aug 28 08:29:14 UTC 2025
Thu Aug 28 08:29:15 UTC 2025
Thu Aug 28 08:29:16 UTC 2025
Thu Aug 28 08:29:17 UTC 2025
Thu Aug 28 08:29:18 UTC 2025
Thu Aug 28 08:29:19 UTC 2025
Thu Aug 28 08:29:20 UTC 2025
Thu Aug 28 08:29:21 UTC 2025
Thu Aug 28 08:29:22 UTC 2025
Thu Aug 28 08:29:23 UTC 2025
Thu Aug 28 08:29:24 UTC 2025
Thu Aug 28 08:29:25 UTC 2025
Thu Aug 28 08:29:26 UTC 2025
Thu Aug 28 08:29:27 UTC 2025
Thu Aug 28 08:29:28 UTC 2025
Thu Aug 28 08:29:29 UTC 2025
Thu Aug 28 08:29:30 UTC 2025
Thu Aug 28 08:29:31 UTC 2025
Thu Aug 28 08:29:32 UTC 2025
Thu Aug 28 08:29:33 UTC 2025
Thu Aug 28 08:29:34 UTC 2025
Thu Aug 28 08:29:35 UTC 2025
Thu Aug 28 08:29:36 UTC 2025
Thu Aug 28 08:29:37 UTC 2025
Thu Aug 28 08:29:38 UTC 2025
Thu Aug 28 08:29:39 UTC 2025
Thu Aug 28 08:29:40 UTC 2025

Thu Aug 28 08:29:41 UTC 2025
Thu Aug 28 08:29:42 UTC 2025
Thu Aug 28 08:29:43 UTC 2025
Thu Aug 28 08:29:44 UTC 2025
Thu Aug 28 08:29:45 UTC 2025
Thu Aug 28 08:29:46 UTC 2025
Thu Aug 28 08:29:47 UTC 2025
Thu Aug 28 08:29:48 UTC 2025
Thu Aug 28 08:29:49 UTC 2025
Thu Aug 28 08:29:50 UTC 2025
Thu Aug 28 08:29:51 UTC 2025
Thu Aug 28 08:29:52 UTC 2025
Thu Aug 28 08:29:53 UTC 2025
Thu Aug 28 08:29:54 UTC 2025
Thu Aug 28 08:29:55 UTC 2025
Thu Aug 28 08:29:56 UTC 2025
Thu Aug 28 08:29:57 UTC 2025
Thu Aug 28 08:29:58 UTC 2025
Thu Aug 28 08:29:59 UTC 2025
Thu Aug 28 08:30:00 UTC 2025
Thu Aug 28 08:30:01 UTC 2025
Thu Aug 28 08:30:02 UTC 2025
Thu Aug 28 08:30:03 UTC 2025
Thu Aug 28 08:30:04 UTC 2025
Thu Aug 28 08:30:05 UTC 2025
Thu Aug 28 08:30:06 UTC 2025
Thu Aug 28 08:30:07 UTC 2025
Thu Aug 28 08:30:08 UTC 2025
Thu Aug 28 08:30:09 UTC 2025
Thu Aug 28 08:30:10 UTC 2025
Thu Aug 28 08:30:11 UTC 2025
Thu Aug 28 08:30:12 UTC 2025
Thu Aug 28 08:30:13 UTC 2025
Thu Aug 28 08:30:14 UTC 2025
Thu Aug 28 08:30:15 UTC 2025
Thu Aug 28 08:30:16 UTC 2025
Thu Aug 28 08:30:17 UTC 2025
Thu Aug 28 08:30:18 UTC 2025

4. Run Busybox

1. Command: `tail -f /dck2/date.log`
 2. Volume to mount: `$(pwd)/sidecar:/dck2`
 3. State: attached
- `docker run -it --rm -v "$(pwd)/sidecar:/dck2" busybox sh`

```
/ # tail -f /dck2/date.log
Thu Aug 28 08:43:14 UTC 2025
Thu Aug 28 08:43:15 UTC 2025
Thu Aug 28 08:43:16 UTC 2025
Thu Aug 28 08:43:17 UTC 2025
Thu Aug 28 08:43:18 UTC 2025
Thu Aug 28 08:43:19 UTC 2025
Thu Aug 28 08:43:20 UTC 2025
Thu Aug 28 08:43:21 UTC 2025
Thu Aug 28 08:43:22 UTC 2025
Thu Aug 28 08:43:23 UTC 2025
Thu Aug 28 08:43:24 UTC 2025
Thu Aug 28 08:43:25 UTC 2025
Thu Aug 28 08:43:26 UTC 2025
Thu Aug 28 08:43:27 UTC 2025
```

5. Check content of `dck2/date.log` with `tail -f`

- `tail -f /dck2/date.log`

6. Exit container

7. Run `docker kill gen_date`

1. Why is the container stoped ?

gen_date tournait avec une boucle infinie (while true; do date >> /dck/date.log; sleep 1; done).

docker kill arrête le processus principal → le conteneur n'a plus rien à exécuter → il s'arrête.

In memory

1. Run busybox with `--tmpfs /test`

- `docker run -it --rm --tmpfs /test busybox`

2. Check with `mount | grep test` that tmpfs is used

- `mount | grep test`

tmpfs on /test type tmpfs (rw,nosuid,nodev,noexec,relatime)