Docker Volume – Complete Deep Dive Tutorial

The Problem: Why Do We Need Volumes?

Containers are:

- Ephemeral
- Have a **Union file system**
- Have a **read-only image** + **writable layer**

When a container is stoped and removed, the writable layer is **destroyed**, so any data stored inside:

```
/var/lib/mysql
/usr/share/nginx/html
/app/logs
```

is **lost**.

To persist data beyond container lifecycle → use volumes or bind mounts

N.B: When containers are stopped - data is retained.

Volume Types in Docker

There are 4:

1. **Anonymous volume**
2. **Named volume**
3. **Bind mount**
4. **tmpfs mount (RAM only)**

Let’s understand each deeply.

Anonymous Volume

Created automatically when a container needs storage

Docker assigns a random name like:

```
d3c9248bc5a8f9301a6e...
```

Use Case

- Short-lived containers
- When you don't care about the volume name
- Automatically cleaned when container is removed (only with `--rm`)

Example

```
docker run -d \
  -v /data \
  --name app1 \
  nginx
```

Here `/data` is mounted as an **anonymous volume**.

View volumes

```
docker volume ls
```

You'll see something like:

```
local     d3c9248bc5a8f9301a6e
```

Named Volume

Explicitly created by the user

- Managed by Docker
- Exists even if container is removed
- Better for production storage

Example – Create a named volume

```
docker volume create mydata
```

Use it in a container

```
docker run -d \
  -v mydata:/var/lib/mysql \
  --name mysql \
  mysql:8
```

Data persists even after destroying container

```
docker rm -f mysql
docker run -d -v mydata:/var/lib/mysql mysql:8
```

The DB still has old data.

Useful for:

- Databases
- Uploads folder
- Application persistent storage

Bind Mount

Host directory is directly mounted inside container

Format:

```
-v /host/path:/container/path
```

Example:

```
docker run -d \
  -v $(pwd)/website:/usr/share/nginx/html \
  -p 8080:80 \
  nginx
```

Changes on your host immediately reflect in container → **useful for development**.

Use Cases

- Local development (auto-reload)
- Logging (map /var/log)
- Config files
- Mapping secrets/certs

Characteristics

- Host controls the directory
- Faster access for dev environments
- Not recommended for production unless needed
- Requires absolute paths

---

tmpfs Mount

Mounts a directory into RAM – not stored in disk

- Very fast
- Data is temporary and disappears when container stops
- Used for:
    - Highly sensitive data
    - Caches
    - Temporary files

Example:

```
docker run -d \
  --mount type=tmpfs,target=/cache \
  nginx
```

Or using `-v` syntax:

```
docker run -d \
  --tmpfs /cache \
  nginx
```

tmpfs Characteristics

- Extremely fast (RAM)
- Not persistent

- Useful for security

Comparison Table

Type	Persistent?	Stored Where?	Use-case	Example
Anonymous Volume	Yes	`/var/lib/docker/volumes/...`	Temporary storage	`-v /data`
Named Volume	Yes	`/var/lib/docker/volumes/mydata/`	DB & persistent storage	`-v mydata:/var/lib/mysql`
Bind Mount	Host directory	Host filesystem	Development, logs	`-v $(pwd):/app`
tmpfs	No (RAM only)	RAM	Cache, security, speed	`--tmpfs /cache`

Where Docker Stores Volume Data? (Important for Interviews)

Default directory:

```
/var/lib/docker/volumes/
```

Inside:

```
mydata/_data/
```

Docker manages the folder, permissions, metadata.

Bind mounts are **NOT stored here** (they point to a host path).

Real-World Use Cases

✔ Named volume

- MySQL, MongoDB, PostgreSQL, Redis
- Media storage
- File uploads

✔ Bind mount

- Live development syncing code
- Configuration injection
- Host logs collection

✔ tmpfs

- Fast temporary storage
- Sensitive security info
- Build caches

✔ Anonymous volume

- Temporary test containers
- Scratch containers
- Disposable services

Examples Using Each Type

---

1. Anonymous Volume Example

```
docker run -d \
  -v /var/log/app \
  myapp
```

Check volume:

```
docker volume ls
```

Inspect:

```
docker volume inspect <id>
```

---

2. Named Volume Example

```
docker volume create mydb

docker run -d \
  -v mydb:/var/lib/mysql \
  mysql:8
```

---

3. Bind Mount Example

Host directory required:

```
mkdir website
echo "Hello" > website/index.html
```

Run:

```
docker run -d \
  -v $(pwd)/website:/usr/share/nginx/html \
  -p 8080:80 \
  nginx
```

Visit:
`http://localhost:8080`

Change file → hot reload.

---

4. tmpfs Example

```
docker run -d \
  --tmpfs /secret \
  busybox sh -c "echo hi > /secret/msg && sleep 1000"
```

Check inside:

```
docker exec -it <id> cat /secret/msg
```

Stop container → data gone.

Final Summary

Feature	Anonymous	Named	Bind Mount	tmpfs
Data persists	✔	✔	On host	No
Managed by Docker	✔	✔	No	✔
Best for DB	No	✔	No	No
Best for Development	No	No	✔	No
Best for temporary secure data	No	No	No	✔