

PERSISTING DATA USING VOLUMES

1. kita dapat meghubungkan volume di host ke container agar bisa saling bertukar data

Persisting Data Using Volumes

Step 1 of 3

data saved inside the container to the `/data` directory should be persisted on the host in the directory `/docker/redis-data`.

```
docker run -v /docker/redis-data:/data \
--name r1 -d redis \
redis-server --appendonly yes ✓
```

We can pipe data into the Redis instance using the following command.

```
cat data | docker exec -i r1 redis-cli --pipe
```

Redis will save this data to disk. On the host we can investigate the mapped direct which should contain the Redis data file.

```
ls /docker/redis-data ✓
```

This same directory can be mounted to a second container. One usage is to have a Docker Container performing backup operations on your data.

Terminal

```
[root@host01 ~]# ls /
bin dev home lib64 mnt proc run srv tmp var
boot etc lib lost+found opt root sbin sys usr
[root@host01 ~]# docker run -v /docker/redis-data:/data \
> --name r1 -d redis \
> redis-server --appendonly yes
bd6ffdebda5d15e97b56323f99c9ad80b24cdd14b370c6d34e5a5aedecb9b389
[root@host01 ~]# cat data | docker exec -i r1 redis-cli --pipe
All data transferred. Waiting for the last reply...
Last reply received from server.
errors: 0, replies: 1
[root@host01 ~]# ls /docker/redis-data
appendonly.aof
[root@host01 ~]# docker ps
CONTAINER ID        PORTS              NAMES              COMMAND              CREATED              STATUS
bd6ffdebda5d        6379/tcp           r1                 "docker-entrypoint.s..." 40 seconds ago      Up 38 se
[root@host01 ~]# docker exec bd6ffdebda5d ls /data
appendonly.aof
[root@host01 ~]#
```

Persisting Data Using Volumes

Step 1 of 3

data saved inside the container to the `/data` directory should be persisted on the host in the directory `/docker/redis-data`.

```
docker run -v /docker/redis-data:/data \
--name r1 -d redis \
redis-server --appendonly yes ✓
```

We can pipe data into the Redis instance using the following command.

```
cat data | docker exec -i r1 redis-cli --pipe
```

Redis will save this data to disk. On the host we can investigate the mapped direct which should contain the Redis data file.

```
ls /docker/redis-data ✓
```

This same directory can be mounted to a second container. One usage is to have a Docker Container performing backup operations on your data.

Terminal

```
[root@host01 ~]# mkdir -p /docker/redis-data/test1
[root@host01 ~]# ls /docker/redis-data/
appendonly.aof test1
[root@host01 ~]# docker exec bd6ffdebda5d ls /data
appendonly.aof
test1
[root@host01 ~]# docker exec bd6ffdebda5d ls /
bin
boot
data
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
```

2. folder yg berada di host dapat di mount lebih dari 1 container secara bersamaan

Persisting Data Using Volumes | Docker | Katacoda - Mozilla Firefox

https://katacoda.com/courses/docker/persisting-data-using-volumes

Persisting Data Using Volumes

Step 1 of 3

We can pipe data into the Redis instance using the following command.

```
cat data | docker exec -i r1 redis-cli --pipe
```

Redis will save this data to disk. On the host we can investigate the mapped direct which should contain the Redis data file.

```
ls /docker/redis-data
```

This same directory can be mounted to a second container. One usage is to have a Docker Container performing backup operations on your data.

```
docker run -v /docker/redis-data:/backup ubuntu ls /backup
```

CONTINUE

Terminal

```
[root@host01 ~]# docker exec bd6ffdebda5d mkdir -p /data/tes3
[root@host01 ~]# docker exec bd6ffdebda5d ls /data
appendonly.aof
tes1
tes2
tes3
[root@host01 ~]# ls /docker/redis-data/
appendonly.aof tes1 tes2 tes3
[root@host01 ~]# docker run -v /docker/redis-data:/backup ubuntu ls /backup
appendonly.aof
tes1
tes2
tes3
[root@host01 ~]#
```

3. dengan opsi `--volume-from` kita dapat mounting volume container lama ke container yg baru tanpa harus mengetahui path data yg di mounting dari host

Persisting Data Using Volumes | Docker | Katacoda - Mozilla Firefox

https://katacoda.com/courses/docker/persisting-data-using-volumes

Persisting Data Using Volumes

Step 2 of 3

Step 2 - Shared Volumes

Data Volumes mapped to the host are great for persisting data. However, to gain access to them from another container you need to know the exact path which can make it error-prone.

An alternate approach is to use `--volumes-from`. The parameter maps the mapped volumes from the source container to the container being launched.

In this case, we're mapping our Redis container's volume to an Ubuntu container. The `/data` directory only exists within our Redis container, however, because of `--volumes-from` our Ubuntu container can access the data.

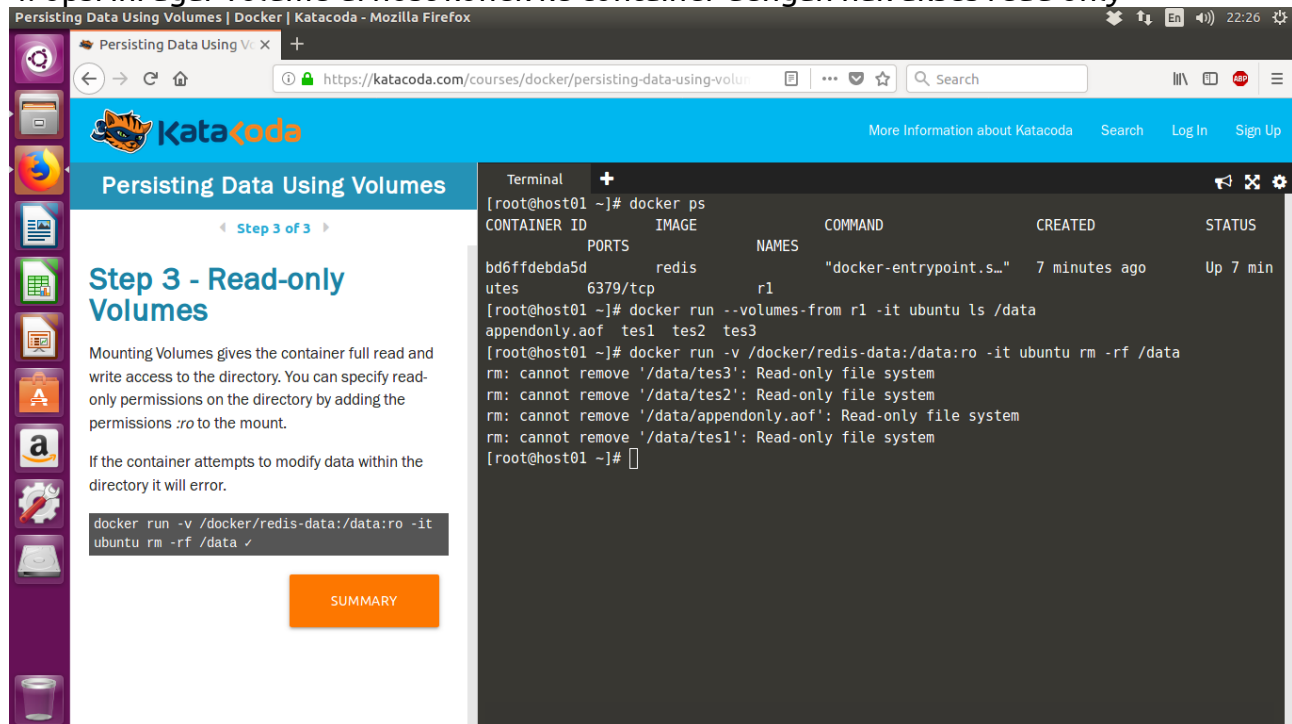
```
docker run --volumes-from r1 -it ubuntu ls /data
```

This allows us to access volumes from other containers without having to be concerned how

Terminal

```
[root@host01 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
bd6ffdebda5d   redis     "docker-entrypoint.s..." 7 minutes ago  Up 7 min
utes
6379/tcp      r1
[root@host01 ~]# docker run --volumes-from r1 -it ubuntu ls /data
appendonly.aof tes1 tes2 tes3
[root@host01 ~]#
```

4. opsi ini agar volume di host konek ke container dengan hak akses read only



Persisting Data Using Volumes

Step 3 - Read-only Volumes

Mounting Volumes gives the container full read and write access to the directory. You can specify read-only permissions on the directory by adding the permissions `:ro` to the mount.

If the container attempts to modify data within the directory it will error.

```
docker run -v /docker/redis-data:/data:ro -it ubuntu rm -rf /data ✓
```

SUMMARY

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
bd6ffdeba5d	redis	"docker-entrypoint.s..."	7 minutes ago	Up 7 min

```
[root@host01 ~]# docker ps
CONTAINER ID   IMAGE    COMMAND                  CREATED       STATUS
bd6ffdeba5d    redis   "docker-entrypoint.s..." 7 minutes ago Up 7 min
utes        6379/tcp    r1

[root@host01 ~]# docker run --volumes-from r1 -it ubuntu ls /data
appendonly.aof tes1 tes2 tes3
[root@host01 ~]# docker run -v /docker/redis-data:/data:ro -it ubuntu rm -rf /data
rm: cannot remove '/data/tes3': Read-only file system
rm: cannot remove '/data/tes2': Read-only file system
rm: cannot remove '/data/appendonly.aof': Read-only file system
rm: cannot remove '/data/tes1': Read-only file system
[root@host01 ~]#
```

dok 06/05/2018 padliyulian@gmail.com