

Volumes

To list your volumes, run `docker volume ls`:

```
hater@hater:~$ docker volume ls
DRIVER      VOLUME NAME
local       desktop_mongodb_data
local       portainer_data
```

To create a new volume, run `docker volume create` and give it a volume name.

```
hater@hater:~$ docker volume create myvolume
myvolume
hater@hater:~$ docker volume ls
DRIVER      VOLUME NAME
local       desktop_mongodb_data
local       myvolume
local       portainer_data
hater@hater:~$
```

To remove a volume, run `docker volume rm` and give it the volume name.

```
hater@hater:~$ docker volume rm myvolume
myvolume
```

Then re-run

our `docker run` command, providing the `-v` mount flag.

```
hater@hater:~$ docker run --rm -d --name apache -p 80:80 -v httpd_htdocs:/usr/local/apache2/htdocs/ httpd:2.4
5efbbca2eea9ea9efc75ab7304a806a6b91695fc7364016f67cbfc32d5545263
hater@hater:~$
```

And re-copy in our modified HTML file.

```
hater@hater:~$ docker cp index.html apache:/usr/local/apache2/htdocs/
```

And run `curl` to verify it worked.

```
hater@hater:~$ curl localhost
<html><body><h1>It works!</h1></body></html>
```

Now to see the volume in action, let's stop the container. By providing the `--rm` flag during run, it should remove the container upon stopping.

```
hater@hater:~$ docker stop apache
apache
```

Then once again start `httpd` with the same run command as last time. This time, however, we can `curl` and see our file changes are still there from before.

```
hater@hater:~$ docker run --rm -d --name apache -p 80:80 -v httpd_htdocs:/usr/local/apache2/htdocs/ httpd:2.4
7eaa018b66f66dae1152b245fb9d26f00d033bf835bc8e6a1efb26c62751b663
hater@hater:~$ curl localhost
<html><body><h1>It works!</h1></body></html>
```

