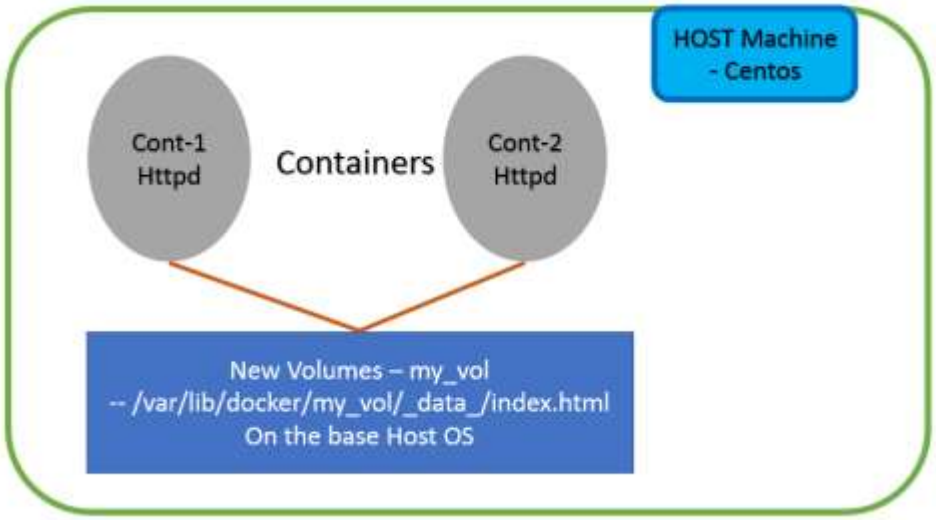
Objective:

Map the container to the storage on the Host machine.

Diagram:



Steps

1. **Create a docker volume**
2. **Choosing the httpd image and understand the problem.**
3. **Creating the container with the external volume**
4. **Sharing the volume between 2 containers.**

Step1:

Create Docker volume

$ docker volume create *<<volume name>>*

```
[root@linux-client ~]# docker volume create my-vol1
my-vol1
[root@linux-client ~]# docker volume ls
DRIVER              VOLUME NAME
local               my-vol1
[root@linux-client ~]#
```

Below is the path where the volumes are created for docker usage.

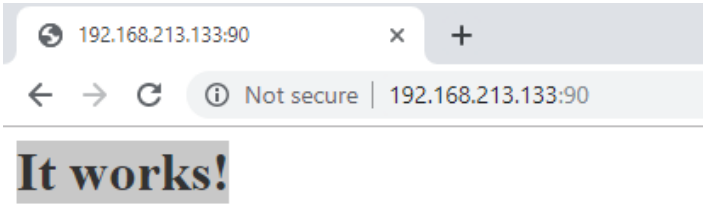
```
[root@linux-client ~]# find / -iname my-vol1
/var/lib/docker/volumes/my-vol1
[root@linux-client ~]#
```

```
[root@linux-client ~]# cd /var/lib/docker/volumes/my-vol1
[root@linux-client my-vol1]# ls
_data
[root@linux-client my-vol1]# cd _data/
[root@linux-client _data]# ls
[root@linux-client _data]# pwd
/var/lib/docker/volumes/my-vol1/_data
[root@linux-client _data]#
```

Step2:

Now, lets consider the httpd image. And lets run a container.

```
[root@linux-client ~]# docker run -d -p 90:80 --name cont01 httpd
2aa65f8723758d71c3ee102688c38046b9caabd5b9fa9a25f792a6cb03c7c94e
```



The content of the above page is in the specific path of the container

```
[root@linux-client ~]# docker exec -it cont01 /bin/bash
root@2aa65f872375:/usr/local/apache2# cd htdocs/
root@2aa65f872375:/usr/local/apache2/htdocs# ls
index.html
root@2aa65f872375:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>It works!</h1></body></html>
root@2aa65f872375:/usr/local/apache2/htdocs#
```

"/usr/local/apache2/htdocs/index.html" → path in the container

Now , if we have to change the content of this file, we need to run the "Dockerfile" everytime and create an custom image.

Which is not practical.

Solution:

We would be mapping the path to the external folder on the base machine with help of "Volumes".

Step3: Creating the container with the external volume

$ docker run -d -p 90:80 -v my-vol1:/usr/local/apache2/htdocs --name cont01 httpd

```
[root@linux-client _data]# docker run -d -p 90:80 -v my-vol1:/usr/local/apache2/htdocs --name cont01 httpd
b571d3f9e2fe676a2c487d39c6267e2f0928c74254b484133562fa3fec7d195a
[root@linux-client _data]# ls
index.html
[root@linux-client _data]# cat index.html
<html><body><h1>It works!</h1></body></html>
[root@linux-client _data]#
```

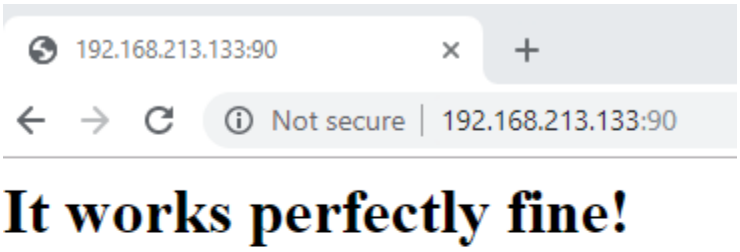
The volume folder on the local machine now has the data what container has.

Now, Edit this "index.html" and add some content.

Content → "It works perfectly fine!"

Save and quit the file → press "esc" & ":wq"

```
[root@linux-client _data]# pwd
/var/lib/docker/volumes/my-vol1/_data
[root@linux-client _data]#
```



Step4: Sharing the volume between 2 containers.

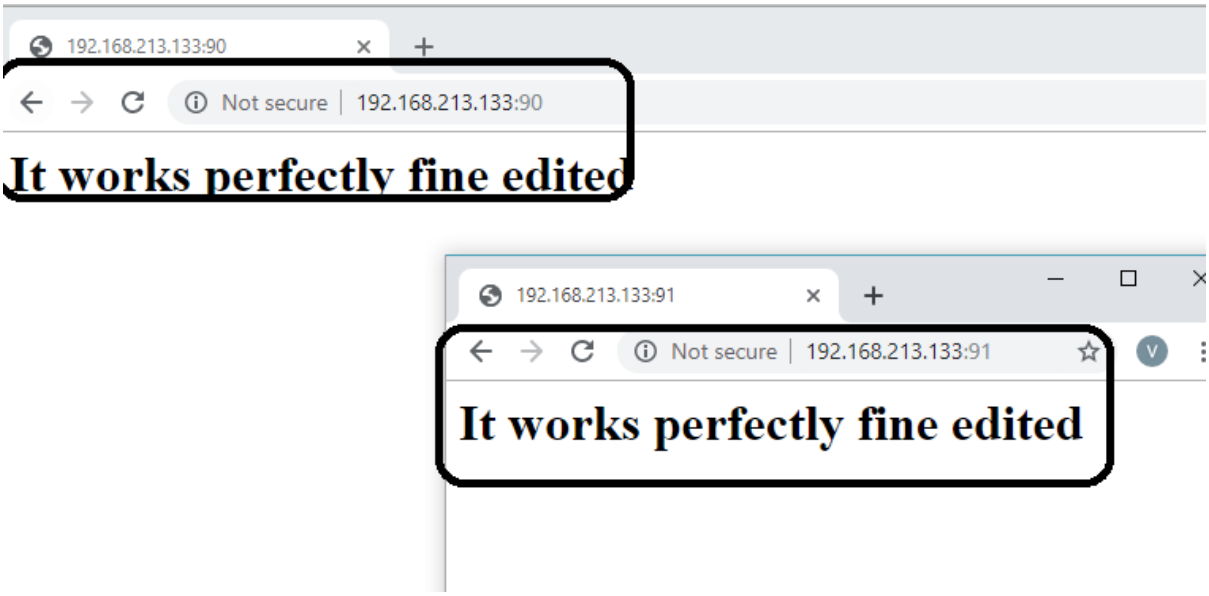
Now, that we are able to control the data in the container, without re-imaging and disturbing the container.

Let's also see how to share the same content to another container.

$ docker run -d -p 91:80 -v my-vol1:/usr/local/apache2/htdocs:ro --name cont02 httpd

```
[root@linux-client _data]# docker ps
CONTAINER ID        IMAGE               COMMAND              CREATED             STATUS              PORTS
 NAMES
c458eabf95b4        httpd               "httpd-foreground"   3 seconds ago       Up 2 seconds        0.0.0.0:90->80/tcp
 cont01
786f45dfa7f7        httpd               "httpd-foreground"   20 seconds ago      Up 19 seconds       0.0.0.0:91->80/tcp
 cont02
[root@linux-client _data]#
```

We have 2 containers running.



Both container showing the same content as its referring to the same content.