

NAME : T MANOJ
EMPID : LYAKE2KHS

Docker Guide: Volume Management, Image Backup, and Container Linking

1. Creating Docker Volumes with Specific Paths

- Create volume with specific dev path using appropriate options
- Inspect the volume to find the mount point location
- Create a container specifying the volume mount
- Verify mounting by creating files on host and checking visibility inside container

```
[root@ip-172-31-88-154 ~]# docker run -d -p "8060:80" --name apachecontainer -v "/docker:/var/www/html" httpd
6bb71cb72c2ff4cd0be43a0d569d8806d33229faee4c158728b86f7276001202
[root@ip-172-31-88-154 ~]#
```

```
[root@ip-172-31-88-154 ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6bb71cb72c2f	httpd	"httpd-foreground"	10 seconds ago	Up 9 seconds	0.0.0.0:8060->80/tcp, :::8060->80/tcp	apachecontainer

← → 🔍 ⚠ Not secure 3.86.83.203:7020 ☆ m ⋮

"hi Cprime"

```
491085415620-6tnyhdsj.us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-0652ee...
aws Search [Alt+S] United States (N. Virginia) cloudprotocols-hybrid-devops (4910-8541-5620) root

7fe19e777c70bc70c0e94005eb38fdb94d9d11f8816d5bd716717fcf0e25148
root@ip-172-31-88-154 ~]# docker exec -it voll /bin/bash
root@97fe19e777c7:/usr/local/apache2# ls
bin build cgi-bin conf error htdocs icons include logs modules
root@97fe19e777c7:/usr/local/apache2# cd htdocs/
root@97fe19e777c7:/usr/local/apache2/htdocs# ls
root@97fe19e777c7:/usr/local/apache2/htdocs# ls
root@97fe19e777c7:/usr/local/apache2/htdocs# exit
exit
root@ip-172-31-88-154 ~]# cd /opt/www/
root@ip-172-31-88-154 www]# ls
root@ip-172-31-88-154 www]# vi index.html
root@ip-172-31-88-154 www]# ll
total 4
-rw-r--r--. 1 root root 12 Apr 10 07:13 index.html
root@ip-172-31-88-154 www]# ls
index.html
root@ip-172-31-88-154 www]# cd
root@ip-172-31-88-154 ~]# cat /opt/www/index.html
hi Cprime"
root@ip-172-31-88-154 ~]#
```

i-0652eeaca28170491 (DOCKER)

PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

2. Working with Local Volumes

- Create local Docker volume
- Determine the volume's mount point through inspection
- Create files in the host location
- Mount volume to container
- Verify files are accessible from within container

```
491085415620-6tnyhdsj.us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?addressFamily=ipv4&connType=standard&instanceId=i-06...
aws Search [Alt+S] United States (N. Virginia) cloudprotocols-hybrid-devops (4910-8541-5620) root

[root@ip-172-31-88-154 ~]# docker volume create myvolume
myvolume
[root@ip-172-31-88-154 ~]# docker inspect myvolume
[
  {
    "CreatedAt": "2025-04-10T07:16:14Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/myvolume/_data",
    "Name": "myvolume",
    "Options": null,
    "Scope": "local"
  }
]
```

i-0652eeaca28170491 (DOCKER)

PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

```
[root@ip-172-31-88-154 _data]# docker run -d -it --name=volume --mount source=myvolume,destination=/usr/local/apache2/htdocs httpd
93bcbac4b5f30a6f47a49f338d7517d86b358faa81db0b502782cd8a83507fcc
[root@ip-172-31-88-154 _data]# docker inspect volume
[
  {
    "Id": "93bcbac4b5f30a6f47a49f338d7517d86b358faa81db0b502782cd8a83507fcc",
    "Created": "2025-04-10T07:30:06.412290442Z",
    "Path": "httpd-foreground",
    "Args": [],
    "State": {
      "Status": "running",
```

```
aws [Search] [Alt+S] United States (N. Virginia) cloudprotocols-hybrid-devops (4910-8541-5620) root

[root@ip-172-31-88-154 ~]# docker exec -it volume /bin/bash
root@93bcbac4b5f3:/usr/local/apache2# ls
bin build cgi-bin conf error htdocs icons include logs modules
root@93bcbac4b5f3:/usr/local/apache2# cd htdocs/
root@93bcbac4b5f3:/usr/local/apache2/htdocs# ls
index.html
root@93bcbac4b5f3:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>It works!</h1></body></html>
root@93bcbac4b5f3:/usr/local/apache2/htdocs# cd
root@93bcbac4b5f3:~# exit
exit
[root@ip-172-31-88-154 ~]# cd /var/lib/docker/volumes/myvolume/_data/
[root@ip-172-31-88-154 _data]# vi index.html
[root@ip-172-31-88-154 _data]# cat index.html
<html><body><h1>Hello Docker</h1></body></html>
[root@ip-172-31-88-154 _data]# cd
[root@ip-172-31-88-154 ~]# docker exec -it volume /bin/bash
root@93bcbac4b5f3:/usr/local/apache2# cd htdocs/
root@93bcbac4b5f3:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>Hello Docker</h1></body></html>
root@93bcbac4b5f3:/usr/local/apache2/htdocs#
```

i-0652eeaca28170491 (DOCKER)

PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

3. Pushing Images to Docker Hub

- Login to Docker Hub using credentials

```
[root@ip-172-31-88-154 _data]# docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/

Username: manoj758
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

- Create a backup of your image before pushing

```
[root@ip-172-31-88-154 ~]# docker tag httpd manoj758/backup:latest
[root@ip-172-31-88-154 ~]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
images	v1.0	2271de943a27	20 hours ago	315MB
image	latest	f440043aac6c	21 hours ago	339MB
images	latest	954f2a896262	21 hours ago	339MB
myimage	latest	0005087706b8	21 hours ago	352MB
<none>	<none>	d15a2aae8801	23 hours ago	248MB
hello	latest	b4f7da17ba23	23 hours ago	339MB
<none>	<none>	d44006bbbf27	25 hours ago	284MB
<none>	<none>	5522d1eea668	25 hours ago	352MB
httpd	latest	10fd72f437c4	2 months ago	148MB
manoj758/backup	latest	10fd72f437c4	2 months ago	148MB

```
[root@ip-172-31-88-154 ~]#
```

i-0652eeaca28170491 (DOCKER)

PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

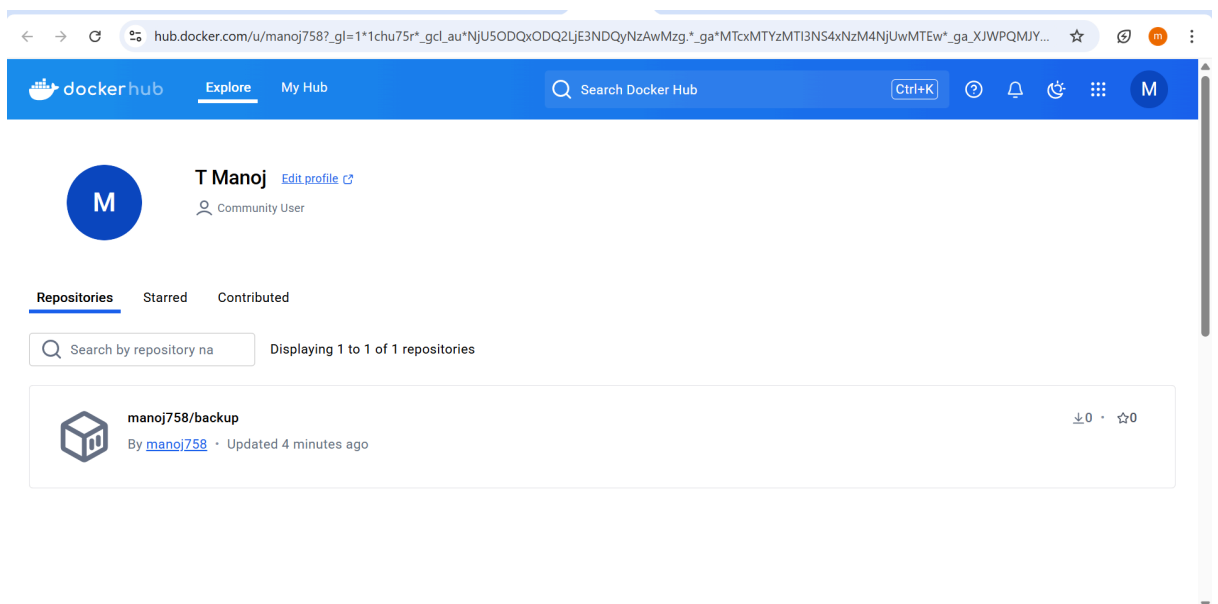
- Tag the image with proper Docker Hub username and repository
- Push the image to Docker Hub

```
aws | Search [Alt+S] | United States (N. Virginia) | cloudprotocols-hybrid-devops (4910-8541-5620) | root

manoj758/backup latest 10fd72f437c4 2 months ago 148MB
[root@ip-172-31-88-154 ~]# docker push manoj758/backup
Using default tag: latest
The push refers to repository [docker.io/manoj758/backup]
8e92d33aa718: Mounted from library/httpd
b8d276594499: Mounted from library/httpd
417809be0bd7: Mounted from library/httpd
5f70bf18a086: Mounted from library/httpd
fe0988bb067f: Mounted from library/httpd
ea680fbff095: Mounted from library/httpd
latest: digest: sha256:2342567c2e013ec1475f76bf62829dd10e6f071b59b8364d43471363eb57bb77 size: 1572
[root@ip-172-31-88-154 ~]# docker push manoj758/backup:latest
The push refers to repository [docker.io/manoj758/backup]
8e92d33aa718: Layer already exists
b8d276594499: Layer already exists
417809be0bd7: Layer already exists
5f70bf18a086: Layer already exists
fe0988bb067f: Layer already exists
ea680fbff095: Layer already exists
latest: digest: sha256:2342567c2e013ec1475f76bf62829dd10e6f071b59b8364d43471363eb57bb77 size: 1572
[root@ip-172-31-88-154 ~]#
```

i-0652eeaca28170491 (DOCKER)
PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

- Confirm successful upload by checking Docker Hub



4. Backing Up and Restoring Containers

- Create a backup image from running container using commit

```
manoj758/backup latest 10fd72f437c4 2 months ago 148MB
[root@ip-172-31-88-154 ~]# docker commit 93bc4b5f3 backup_image
sha256:1dc0e17c13a0f4e32525833682e8dea3d212aa95eac4a1def869279165c81d5c
[root@ip-172-31-88-154 ~]#
```

```
[root@ip-172-31-88-154 ~]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
backup_image        latest   1dc0e17c13a0   18 seconds ago 148MB
```

- Export the image as a tar/zip file

```
aws
[Search] [Alt+S] [Icons] [Settings] United States (N. Virginia) cloudprotocols-hybrid-devops (4910-8541-5620) root

[root@ip-172-31-88-154 ~]# docker save -o ~/container.tar backup_image
[root@ip-172-31-88-154 ~]# docker rmi backup_image
Untagged: backup_image:latest
Deleted: sha256:1dc0e17c13a0f4e32525833682e8dea3d212aa95eac4aldef869279165c81d5c
Deleted: sha256:1266ca07f4ebc1f22bd60c8ab383e9da0a27b17e5931c78f2adcfd8675d76c5
[root@ip-172-31-88-154 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
images              vl.0               2271de943a27       21 hours ago       315MB
image               latest            f440043aac6c       21 hours ago       339MB
images              latest            954f2a896262       21 hours ago       339MB
myimage             latest            0005087706b8       21 hours ago       352MB
<none>              <none>            d15a2aae8801       23 hours ago       248MB
hello               latest            b4f7da17ba23       23 hours ago       339MB
<none>              <none>            dd4006bbbf27       25 hours ago       284MB
<none>              <none>            5522d1eea668       25 hours ago       352MB
manoj758/backup     latest            10fd72f437c4       2 months ago       148MB
httpd               latest            10fd72f437c4       2 months ago       148MB
[root@ip-172-31-88-154 ~]# docker load -i container
error during connect: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/images/load?quiet=0": read container: is a directory
[root@ip-172-31-88-154 ~]# docker load -i container.tar
d25eba47854f: Loading layer [=====>] 5.632kB/5.632kB
```

i-0652eeaca28170491 (DOCKER)

- Delete original image to test restore process
- Load the image back from the tar file
- Verify image is properly restored

```
[root@ip-172-31-88-154 ~]# docker load -i container.tar
d25eba47854f: Loading layer [=====>] 5.632kB/5.632kB
Loaded image: backup_image:latest
[root@ip-172-31-88-154 ~]# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
backup_image         latest            1dc0e17c13a0       2 minutes ago       148MB
images              vl.0               2271de943a27       21 hours ago       315MB
image               latest            f440043aac6c       21 hours ago       339MB
images              latest            954f2a896262       21 hours ago       339MB
myimage             latest            0005087706b8       21 hours ago       352MB
<none>              <none>            d15a2aae8801       23 hours ago       248MB
hello               latest            b4f7da17ba23       23 hours ago       339MB
<none>              <none>            dd4006bbbf27       25 hours ago       284MB
<none>              <none>            5522d1eea668       25 hours ago       352MB
httpd               latest            10fd72f437c4       2 months ago       148MB
manoj758/backup     latest            10fd72f437c4       2 months ago       148MB
[root@ip-172-31-88-154 ~]#
```

5. Connecting Different Containers

- Create containers that need to communicate

```
aws
[Search] [Alt+S] [Icons] [Settings] United States (N. Virginia) cloudprotocols-hybrid-devops (4910-8541-5620) root

[root@ip-172-31-88-154 ~]# docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql
ceal72a6e83b: Pull complete
daac2c594bdd: Pull complete
cb8acbfe2440c: Pull complete
fae51f7de1fb: Pull complete
b2ead3e96e6b: Pull complete
769c3ac51f88: Pull complete
79f239a40e62: Pull complete
c11056354384: Pull complete
49978e7ccddf: Pull complete
548990e33276: Pull complete
Digest: sha256:0596fa224cdf3b3355ce3ddbdf7ce77be27ec9e51841dfc5d2e1c8b81eea69d2
Status: Downloaded newer image for mysql:latest
docker.io/library/mysql:latest
[root@ip-172-31-88-154 ~]#
```

i-0652eeaca28170491 (DOCKER)

PublicIPs: 3.86.83.203 PrivateIPs: 172.31.88.154

```

[root@ip-172-31-88-154 ~]# docker run -d --name db -e MYSQL_ROOT_PASSWORD=admin mysql
0c84ab69f2d06e37d900c206a3bec6061eafe0db53851dea344b979e6bee1912
[root@ip-172-31-88-154 ~]#

```

- Use proper networking options to link containers

```

[ec2-user@ip-172-31-94-95 ~]$ docker run -d --name webserver --link db:mysql -p
'8010:80" wordpress
Unable to find image 'wordpress:latest' locally
latest: Pulling from library/wordpress
9a628cdd7ccc: Already exists
0a85361c459c: Pull complete
05de80987da0: Pull complete
0990af6e132e: Pull complete
1eb9bf28dacd: Pull complete
c4d2115b4d8e: Pull complete
0ff907e301b0: Pull complete
c94d5002a272: Pull complete
51fd3dcac87d: Pull complete
dbd03e28478e: Pull complete
9d6b76948f51: Pull complete
054cc9f6db6e: Pull complete
eb505bfebc59: Pull complete
4f4fb700ef54: Pull complete
f39d7ba7e8eb: Pull complete
754471b71d01: Pull complete
11232bf31512: Pull complete

```

- Example: Create WordPress container linking to database container
- Verify connection by accessing application on assigned port

