

Bind Mount

What is Bind Mount?

Bind mounting is the process of mounting a file or folder stored anywhere on the host file system into the running container.

How to Bind Mount in Docker?

1. To understand bind mounting in Docker, create a privileged container as shown below. The tag “**--privileged = true**” turns off all the protection and securities, as we have to manipulate the file system.

```
docker run -ti --rm --privileged=true ubuntu bash
```

2. Inside the container, create a directory named “example” and move into the directory.

```
root@2b1400fdb879:/# mkdir example
root@2b1400fdb879:/# cd example
```

3. Inside the example directory, create another directory named “first”. Get into the first directory and create few files such as a, b, c, d, e, and f using the touch command.

```
fdb879:/example# mkdir first
fdb879:/example# cd first
fdb879:/example/first# touch a b c d e f
```

4. Get back to the example directory and create another directory named “second”. Move inside the second directory, create files prefixed with second such as second-a, second-b, second-c and second-d.

```
fdb879:/example/first# cd ..
fdb879:/example# mkdir second
fdb879:/example# cd second
fdb879:/example/second# touch second-a second-b second-c second-d
```

5. In the example directory, use the command "ls -R" to see the directory and sub-files structure.

```
root@2b1400fdb879:/example# ls -R
.:
first second

./first:
a b c d e f

./second:
second-a second-b second-c second-d
root@2b1400fdb879:/example#
```

6. In the following command, "**mount -o bind second first**". The content of the second directory will be placed over the top of the first directory. However, the content of the first directory is not deleted, it is covered up.

```
root@2b1400fdb879:/example# mount -o bind second first
root@2b1400fdb879:/example# ls -R
.:
first second

./first:
second-a second-b second-c second-d

./second:
second-a second-b second-c second-d
root@2b1400fdb879:/example#
```

7. To uncover the bind mount and recover the initial files of first directory, use the command **"umount first"**. The file structure will be restored back to the initial structure.

```
root@2b1400fdb879:/example# umount first
root@2b1400fdb879:/example# ls -R
.:
first second

./first:
a b c d e f

./second:
second-a second-b second-c second-d
root@2b1400fdb879:/example#
```