VFS storage driver

- Configure Docker with the vfs storage driver
- How the vfs storage driver works
  - Example: Image and container on-disk constructs
- Related information

The VFS storage driver isn't a union filesystem. Each layer is a directory on disk, and there is no copy-on-write support. To create a new layer, a "deep copy" is done of the previous layer. This leads to lower performance and more space used on disk than other storage drivers. However, it is robust, stable, and works in every environment. It can also be used as a mechanism to verify other storage back-ends against, in a testing environment.

Configure Docker with the `vfs` storage driver

1. Stop Docker.

   ```
   $ sudo systemctl stop docker
   ```

2. Edit `/etc/docker/daemon.json`. If it doesn't yet exist, create it. Assuming that the file was empty, add the following contents.

   ```
   {
     "storage-driver": "vfs"
   }
   ```

   If you want to set a quota to control the maximum size the VFS storage driver can use, set the `size` option on the `storage-opts` key.

   ```
   {
     "storage-driver": "vfs",
     "storage-opts": ["size=256M"]
   ```

```
}
```

Docker doesn't start if the `daemon.json` file contains invalid JSON.

3. Start Docker.

```
$ sudo systemctl start docker
```

4. Verify that the daemon is using the `vfs` storage driver. Use the `docker info` command and look for `Storage Driver`.

```
$ docker info

Storage Driver: vfs
...
```

Docker is now using the `vfs` storage driver. Docker has automatically created the `/var/lib/docker/vfs/` directory, which contains all the layers used by running containers.

How the `vfs` storage driver works

Each image layer and the writable container layer are represented on the Docker host as subdirectories within `/var/lib/docker/`. The union mount provides the unified view of all layers. The directory names don't directly correspond to the IDs of the layers themselves.

VFS doesn't support copy-on-write (COW). Each time a new layer is created, it's a deep copy of its parent layer. These layers are all located under `/var/lib/docker/vfs/dir/`.

Example: Image and container on-disk constructs

The following `docker pull` command shows a Docker host downloading a Docker image comprising five layers.

```
$ docker pull ubuntu

Using default tag: latest
latest: Pulling from library/ubuntu
e0a742c2abfd: Pull complete
486cb8339a27: Pull complete
```

```
dc6f0d824617: Pull complete
4f7a5649a30e: Pull complete
672363445ad2: Pull complete
Digest: sha256:84c334414e2bfdcae99509a6add166bbb4fa4041dc3fa6af08046a66fed3005f
Status: Downloaded newer image for ubuntu:latest
```

After pulling, each of these layers is represented as a subdirectory of `/var/lib/docker/vfs/dir/`. The directory names do not correlate with the image layer IDs shown in the `docker pull` command. To see the size taken up on disk by each layer, you can use the `du -sh` command, which gives the size as a human-readable value.

```
$ ls -l /var/lib/docker/vfs/dir/

total 0
drwxr-xr-x.  2 root root  19 Aug  2 18:19
3262dfbe53dac3e1ab7dcc8ad5d8c4d586a11d2ac3c4234892e34bff7f6b821e
drwxr-xr-x. 21 root root 224 Aug  2 18:23
6af21814449345f55d88c403e66564faad965d6afa84b294ae6e740c9ded2561
drwxr-xr-x. 21 root root 224 Aug  2 18:23
6d3be4585ba32f9f5cbff0110e8d07aea5f5b9fbb1439677c27e7dfee263171c
drwxr-xr-x. 21 root root 224 Aug  2 18:23
9ecd2d88ca177413ab89f987e1507325285a7418fc76d0dcb4bc021447ba2bab
drwxr-xr-x. 21 root root 224 Aug  2 18:23
a292ac6341a65bf3a5da7b7c251e19de1294bd2ec32828de621d41c7ad31f895
drwxr-xr-x. 21 root root 224 Aug  2 18:23
e92be7a4a4e3ccbb7dd87695bca1a0ea373d4f673f455491b1342b33ed91446b
```

```
$ du -sh /var/lib/docker/vfs/dir/*

4.0K□/var/lib/docker/vfs/dir/3262dfbe53dac3e1ab7dcc8ad5d8c4d586a11d2ac3c4234892e34bff7f6b821e
125M□/var/lib/docker/vfs/dir/6af21814449345f55d88c403e66564faad965d6afa84b294ae6e740c9ded2561
104M□/var/lib/docker/vfs/dir/6d3be4585ba32f9f5cbff0110e8d07aea5f5b9fbb1439677c27e7dfee263171c
125M□/var/lib/docker/vfs/dir/9ecd2d88ca177413ab89f987e1507325285a7418fc76d0dcb4bc021447ba2bab
104M□/var/lib/docker/vfs/dir/a292ac6341a65bf3a5da7b7c251e19de1294bd2ec32828de621d41c7ad31f895
104M□/var/lib/docker/vfs/dir/e92be7a4a4e3ccbb7dd87695bca1a0ea373d4f673f455491b1342b33ed91446b
```

The above output shows that three layers each take 104M and two take 125M. These directories have only small differences from each other, but they all consume the same amount of disk space. This is one of the disadvantages of using the `vfs` storage driver.

Related information

- Understand images, containers, and storage drivers
- Select a storage driver

□□□ #2
□ □□□ □□□ 2026-10-31 19:30:54 UTC
□ □□□ □□□ 2026-10-31 19:30:55 UTC