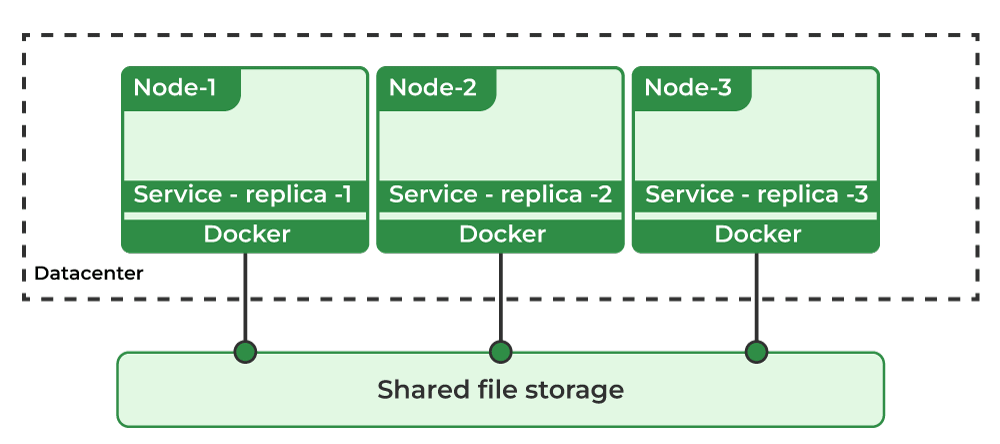
**Phần 17 Gắn một Volume bên trong Docker Container**

Khi bạn làm việc trên kiến ​​trúc dịch vụ vi mô sử dụng các container Docker, bạn tạo nhiều container Docker để tạo và kiểm tra các thành phần khác nhau của ứng dụng. Bây giờ, một số thành phần đó có thể yêu cầu chia sẻ tệp và thư mục. Nếu bạn sao chép cùng một tệp trong tất cả các container riêng biệt, điều này có thể dẫn đến việc tăng kích thước hình ảnh không cần thiết và việc thay đổi tệp trong một container sẽ không tạo ra thay đổi tương ứng trong cùng một tệp trong các container khác.



Do đó, bạn cần một thư mục hoặc ổ đĩa được chia sẻ mà bạn có thể gắn kết trên nhiều [Docker Container](https://www.geeksforgeeks.org/containerization-using-docker) và tất cả chúng đều có quyền truy cập được chia sẻ vào một tệp hoặc thư mục cụ thể. Docker cho phép bạn gắn kết các ổ đĩa được chia sẻ trong nhiều container. Trong bài viết này, chúng ta sẽ gắn kết một ổ đĩa vào các container khác nhau và kiểm tra xem các thay đổi trong tệp có được chia sẻ giữa tất cả các container hay không.

**Volumes**

Volume trong docker là cách được ưa chuộng để triển khai ứng dụng stateful set dưới dạng container. Bạn có thể quản lý và lưu trữ dữ liệu docker bên ngoài vòng đời docker. Volume được docker quản lý hoàn toàn, không giống như bind mount. Volume Docker sẽ cho phép bạn chia sẻ dữ liệu giữa các container docker và máy chủ docker. Có hai loại volume trong docker. Bạn [lưu trữ dữ liệu trong Docker.](https://https/www.geeksforgeeks.org/data-storage-in-docker)

* Các tập sách có tên.
* Liên kết các tập.

**Tạo và quản lý khối lượng | Hướng dẫn từng bước**

Để biết thêm về các lệnh Docker, hãy tham khảo [Docker – Lệnh hướng dẫn.](https://www.geeksforgeeks.org/docker-instruction-commands)

**Bước 1: Hiển thị tất cả các Docker Volume hiện có**

Để hiển thị tất cả các [Docker Volume hiện có,](https://www.geeksforgeeks.org/what-is-docker-volume/) bạn có thể sử dụng lệnh list như sau.

**sudo docker volume ls**

**Bước 2: Tạo một khối lượng**

Để tạo một Docker Volume mới, bạn có thể sử dụng Lệnh Volume Create.

**sudo docker volume tạo geeksforgeeks**

**Bước 3: Kiểm tra Docker Volumes**

Để biết thông tin chi tiết về các Ổ đĩa bạn đã tạo, bạn có thể sử dụng Lệnh Kiểm tra Ổ đĩa.

**sudo docker kiểm tra khối lượng geeksforgeeks**

**Bước 4: Gắn Docker Volume**

Sau khi tạo Volume, bước tiếp theo là gắn Volume vào Docker Container. Chúng ta sẽ tạo Docker Container với Ubuntu base Image mà bạn sẽ đề cập trong [Dockerfile](https://www.geeksforgeeks.org/docker-concept-of-dockerfile) và gắn Volume ***geeksforgeeks*** vào Container đó bằng cờ -v. Bạn có thể [cài đặt gói Linux](https://www.geeksforgeeks.org/how-to-install-linux-packages-inside-a-docker-container) bằng cách sử dụng Dockerfile.

**sudo docker run -it -v geeksforgeeks:/shared-volume --name my-container-01 ubuntu**

Lệnh trên gắn ổ đĩa ***geeksforgeeks*** vào thư mục có tên là **shared-volume** bên trong Docker Container có tên là **my-container-01.**

**Bước 5: Tạo một tệp bên trong Docker Volume**

 Bên trong bash của Container, tạo một tệp mới và thêm một số nội dung.

ls   
cd /shared-volume   
echo “GeeksforGeeks” > geeksforgeeks.txt   
ls   
exit

**Bước 6: Tạo một Container khác và Mount Volume**

Tạo một Docker Container khác có tên là **my-container-02 geeks cho geeks** bên trong Container.

**sudo docker run -it -v geeksforgeeks:/shared-volume --name my-container-02 ubuntu**

Nếu bạn vào thư mục **shared-volume** và liệt kê các tệp, bạn sẽ tìm thấy tệp **geeksforgeeks.txt** mà bạn đã tạo trong cùng một ổ đĩa nhưng được gắn kết trong **my-container-01** trước đó và nó cũng có cùng nội dung bên trong. Điều này là do ổ đĩa được chia sẻ giữa hai Container.

**Bước 7: Xóa một ổ đĩa**

Trước khi xóa ổ đĩa, bạn nên đảm bảo rằng bạn đã sao lưu ổ đĩa vì nó sẽ xóa vĩnh viễn dữ liệu được lưu trữ trong đó. Bằng cách sử dụng lệnh sau, bạn có thể xóa ổ đĩa docker. Để biết thêm các lệnh trong Docker, hãy tham khảo Docker Commands.

**docker volume rm volume\_name\_or\_id**

Thay vì volume\_name\_ hoặc \_id, bạn nên sử dụng tên ổ đĩa và id ổ đĩa của mình.

**Bước 8: Bắt đầu một Container với một Volume**

Trước khi bắt đầu một container với sự trợ giúp của volume, trước tiên bạn nên tạo volume. Sau đó, bạn sử dụng lệnh sau để tạo một container. Trước khi tạo một container, bạn cần một [Docker image.](https://www.geeksforgeeks.org/what-is-docker-images)

**docker run -d --name mongo -v mongovol:/data/db -e MONGO\_INITDB\_ROOT\_USERNAME=devdb -e MONGO\_INITDB\_ROOT\_PASSWORD=devdb123**

**“-v mongovol:/data/db”**biểu thị tên của ổ đĩa và “ **/data/db”** biểu thị thư mục mà ổ đĩa sẽ được gắn kết. **“-e”** biểu thị biến môi trường do chính các nhà phát triển viết, có thể là **“tên người dùng và mật khẩu”.**

**Bước 9: Sử dụng Volume với Docker Compose**

Trong khi ghi tệp [docker-compose,](https://www.geeksforgeeks.org/docker-compose) bạn có thể gắn ổ đĩa được tạo để lưu trữ dữ liệu như hiển thị sau.

* **services:** Phần này định nghĩa các dịch vụ hoặc container tạo nên ứng dụng của bạn. Trong trường hợp này, có một dịch vụ có tên là “ [frontend](https://www.geeksforgeeks.org/frontend-vs-backend/) ” .
* **frontend:** Đây là tên của dịch vụ. Đây có thể là bất kỳ tên nào bạn chọn để đại diện cho vùng chứa này trong ứng dụng của bạn.
* **image:** node:lts: Chỉ định hình ảnh Docker được sử dụng để tạo vùng chứa. Ở đây, nó sử dụng hình ảnh Node.js chính thức được gắn thẻ “lts” (Hỗ trợ dài hạn), thường bao gồm phiên bản LTS mới nhất của Node.js.
* **volumes:** Tiểu mục này chỉ định bất kỳ ổ đĩa nào cần được gắn vào vùng chứa. Các ổ đĩa cho phép [lưu trữ dữ liệu liên tục](https://www.geeksforgeeks.org/what-is-persistent-storage/) ngay cả khi vùng chứa bị xóa.
* **- mongovol:/data/db:** Dòng này định nghĩa volume có tên “mongovol” sẽ được gắn vào container tại đường dẫn “/data/db”. Ở đây, “/data/db” có thể biểu thị thư mục dữ liệu mặc định cho một [container MongoDB.](https://www.geeksforgeeks.org/how-to-run-mongodb-as-a-docker-container/) Việc gắn volume này đảm bảo rằng dữ liệu được lưu trữ trong container tại “/data/db” được lưu trữ bên ngoài vòng đời của container.
* **volumes:** Phần này định nghĩa các ổ đĩa được đặt tên có thể được sử dụng bởi các dịch vụ được xác định ở trên.
* **mongovol:** Dòng này định nghĩa một volume được đặt tên là “mongovol.” Volume được đặt tên cung cấp một cách để lưu trữ dữ liệu vượt quá thời gian tồn tại của một container duy nhất. Volume này có thể được chia sẻ giữa nhiều container hoặc dịch vụ, cho phép dữ liệu được chia sẻ và lưu trữ trên các container hoặc dịch vụ đó.

Trong tệp soạn thảo ở trên, bạn có thể thấy cột “ **volume”.** Trong khu vực đó, bạn nên đề cập đến tên của volume và thư mục hoặc đường dẫn của volume mà bạn nên đề cập.

**Sự khác biệt giữa hành vi -v và –mount**

Các tùy chọn -vvà --mounttrong Docker đều có mục đích gắn các ổ đĩa vào vùng chứa, nhưng chúng có một số khác biệt chính về hành vi và khả năng:

1. **Cú pháp:**
   * **-v** : -vTùy chọn là cú pháp viết tắt để chỉ định các tệp đính kèm ổ đĩa. Nó lấy hai đối số: tên ổ đĩa hoặc đường dẫn thư mục lưu trữ và đường dẫn thư mục chứa.
   * **–mount** : --mountTùy chọn này cung cấp một cách rõ ràng và linh hoạt hơn để gắn các ổ đĩa. Nó cho phép kiểm soát tốt hơn các tùy chọn ổ đĩa và có thể được sử dụng để gắn các ổ đĩa từ nhiều nguồn khác nhau.
2. **Khả năng:**
   * **-v** : -vTùy chọn này đơn giản hơn và phù hợp với việc gắn ổ đĩa cơ bản. Nó hỗ trợ việc gắn ổ đĩa từ máy chủ cục bộ hoặc ổ đĩa được đặt tên được tạo bằng docker volume create.
   * **–mount** : --mountTùy chọn này linh hoạt và mạnh mẽ hơn. Nó cho phép chỉ định nhiều tùy chọn khác nhau như trình điều khiển ổ đĩa, mount chỉ đọc, truyền bá mount, v.v. Tùy chọn này có thể mount các ổ đĩa từ nhiều nguồn khác nhau như thư mục cục bộ, ổ đĩa được đặt tên hoặc dịch vụ ổ đĩa từ xa như [AWS EBS](https://www.geeksforgeeks.org/introduction-to-aws-elastic-block-storeebs/) hoặc [Azure File Storage.](https://www.geeksforgeeks.org/microsoft-azure-map-a-network-file-share-to-azure-windows-vm/)
3. **Tùy chọn âm lượng:**
   * **-v** : Với -vtùy chọn này, bạn chỉ có thể chỉ định các tùy chọn cơ bản cho ổ đĩa như gắn kết chỉ đọc và hành vi lưu trữ đệm.
   * **–mount** : --mountTùy chọn này cung cấp nhiều tùy chọn ổ đĩa hơn, bao gồm chỉ định trình điều khiển ổ đĩa, gắn kết chỉ đọc, chế độ truyền bá gắn kết và các tùy chọn nâng cao hơn như chỉ định nhãn ổ đĩa và thiết lập quyền ổ đĩa tùy chỉnh.
4. **Uyển chuyển:**
   * **-v** : -vTùy chọn này ngắn gọn hơn và dễ sử dụng hơn cho các tác vụ gắn ổ đĩa đơn giản.
   * **–mount** : --mountTùy chọn này cung cấp nhiều tính linh hoạt và khả năng kiểm soát hơn, phù hợp với các tình huống gắn ổ đĩa phức tạp khi yêu cầu cấu hình cụ thể.

**Điền vào một khối lượng bằng cách sử dụng một container**

Bạn có thể điền thông tin vào ổ đĩa bằng cách làm theo các bước dưới đây.

**Bước 1:** Tạo một ổ đĩa docker.

docker volume tạo my\_volume

**Bước 2:** Tạo một vùng chứa để chứa dữ liệu vào ổ đĩa.

docker run -d --name mongo -v mongovol:/data/db   
-e MONGO\_INITDB\_ROOT\_USERNAME=devdb MONGO\_INITDB\_ROOT\_PASSWORD=devdb123

**Bước 3:** Kiểm tra ổ đĩa đã được điền đầy đủ thông tin.

docker chạy --rm -v mongovol:/dữ liệu/db busybox ls /dữ liệu/db

**Chia sẻ dữ liệu giữa các máy**

Khi triển khai ứng dụng chịu lỗi, bạn cần cấp quyền cho nhiều bản sao của cùng một dịch vụ vào cùng một tệp để bạn có thể sử dụng một số dịch vụ như [S3](https://www.geeksforgeeks.org/introduction-to-aws-simple-storage-service-aws-s3) , [NFS](https://www.geeksforgeeks.org/network-file-system-nfs) , v.v. như thể hiện trong hình sau.

**Sử dụng trình điều khiển âm lượng**

Bạn có thể tạo các khối lượng docker bằng cách sử dụng lệnh sau trước khi tạo một container. Nếu bạn muốn tạo một khối lượng trong khi tạo container, bạn có thể thực hiện điều đó bằng cách sử dụng các trình điều khiển khối lượng trong docker. Bạn có thể tạo một khối lượng bằng cách sử dụng trình điều khiển khối lượng như được hiển thị trong hình sau.

docker volume create --driver vieux/sshfs <tên vlome bạn muốn tạo>

**Tạo một dịch vụ tạo ra một ổ đĩa NFS**

Bạn có thể tạo ổ đĩa NFS trong khi tạo dịch vụ như được đề cập bên dưới.

phiên bản: '3'   
dịch vụ:   
 nfs-server:   
 hình ảnh: tomact:mới nhất   
 tên\_container: nfs\_container   
 khối lượng:   
 - /data:/nfs\_share   
  
 nfs-client:   
 hình ảnh: busybox   
 tên\_container: nfs\_client   
 lệnh: tail -f /dev/null   
 khối lượng:   
 - nfs-data:/data   
  
khối lượng:   
 nfs-data:   
 driver\_opts:   
 loại: nfs

* **Máy chủ NFS:**
  + **Hình ảnh:** Sử dụng hình ảnh có tên “ [tomact](https://www.geeksforgeeks.org/how-to-start-tomcat-server/) ” (có thể là lỗi đánh máy của “tomcat”) với phiên bản mới nhất.
  + **Tên vùng chứa:** Đặt tên của vùng chứa đang chạy máy chủ NFS thành “nfs\_container”.
  + **Khối lượng:** Ánh xạ thư mục máy chủ “/data” sang thư mục chứa “/nfs\_share”, cho phép chia sẻ dữ liệu giữa máy chủ và chứa.
* **Máy khách NFS:**
  + **Hình ảnh:** Sử dụng hình ảnh “busybox”, cung cấp môi trường Linux tối thiểu.
  + **Tên vùng chứa:** Đặt tên của vùng chứa đang chạy máy khách NFS thành “nfs\_client”.
  + **Lệnh:** Chỉ định lệnh để duy trì việc chạy container (“tail -f /dev/null”).
  + **Volumes:** Tạo một volume có tên là “nfs-data” và gắn nó vào thư mục “/data” bên trong container. Volume này sử dụng trình điều khiển NFS để chia sẻ dữ liệu với máy chủ NFS.
* **Tập:**
  + **nfs-data:** Xác định một ổ đĩa được đặt tên là “nfs-data”.
  + **Tùy chọn trình điều khiển:** Chỉ định tùy chọn cho trình điều khiển ổ đĩa, đặt loại thành “nfs”. Điều này cho biết ổ đĩa sẽ được quản lý bởi trình điều khiển [NFS (Hệ thống tệp mạng)](https://www.geeksforgeeks.org/network-file-system-nfs/) , cho phép vùng chứa truy cập các tệp được lưu trữ từ xa trên máy chủ NFS.

**Cách sử dụng Docker Volumes để duy trì các thay đổi**

Sử dụng Docker volumes là một kỹ thuật cơ bản để duy trì các thay đổi được thực hiện trong container. Sau đây là cách bạn có thể thực hiện:

* **Tạo một Volume:** Đầu tiên, tạo một Docker volume bằng docker volume createlệnh. Ví dụ:

docker volume tạo my\_volume

* **Gắn Volume vào Container:** Khi chạy container, hãy sử dụng -vcờ để gắn volume vào một thư mục cụ thể trong container. Ví dụ:

docker chạy -d -v my\_volume:/ứng dụng my\_image

* **Thực hiện thay đổi bên trong Container:** Truy cập container đang chạy bằng docker exec -it <container\_id> /bin/bashhoặc các lệnh tương tự. Sau đó, thực hiện thay đổi đối với các tệp hoặc thư mục trong ổ đĩa được gắn kết.
* **Xác minh tính bền bỉ:** Thoát khỏi vùng chứa và khởi động một phiên bản khác sử dụng cùng một ổ đĩa. Xác minh rằng các thay đổi được thực hiện trong vùng chứa trước đó được duy trì. Ví dụ:

docker chạy -d -v my\_volume:/ứng dụng my\_image