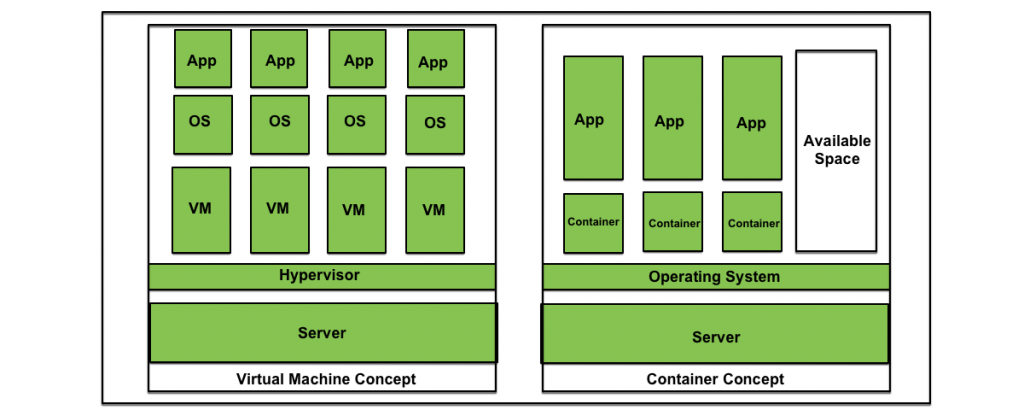
**Phần 13: Ảo hóa với Docker Container**

**1. Container được xây dựng như thế nào bằng công nghệ ảo hóa?**

Để trả lời câu hỏi container cho phép ảo hóa như thế nào , chúng ta hãy thảo luận chi tiết về ảo hóa là gì và container tham gia vào khái niệm này như thế nào. Ảo hóa là một kỹ thuật, về cơ bản tạo ra ảo ảnh về một tài nguyên như máy tính để bàn, bộ lưu trữ, mạng hoặc hệ điều hành. Thiết bị, ứng dụng và người dùng có khả năng tương tác với các tài nguyên này. Ảo ảnh này còn được gọi là ảo hóa mở rộng khả năng của các hệ thống truyền thống, vốn bị giới hạn bởi tài nguyên vật lý của chính chúng. Bây giờ, container cho phép ảo hóa này cho các ứng dụng được triển khai trong đó. Các ứng dụng trong container chạy độc lập, tách biệt khỏi bất kỳ tài nguyên vật lý nào. Container ảo hóa hệ điều hành, CPU, bộ nhớ, bộ lưu trữ và tài nguyên mạng, do đó cung cấp một môi trường được kiểm soát có thể tăng hoặc giảm quy mô theo yêu cầu. Container cũng đóng gói ứng dụng cùng với các phụ thuộc và tệp cần thiết, cho phép triển khai ứng dụng trên bất kỳ môi trường nào mà không cần phải cấu hình máy chủ, phần cứng hoặc phần mềm. Điều này vẫn nghe giống với ảo hóa được triển khai thông qua máy ảo phải không? Vì vậy, trước khi tìm hiểu thêm về những gì container có thể làm, chúng ta hãy làm rõ container khác với máy ảo như thế nào.

**2. Họ cung cấp giải pháp thay thế cho máy ảo như thế nào ?**

Máy [ảo](https://www.geeksforgeeks.org/types-of-virtual-machines/) là một dạng ảo hóa phần cứng. Phần cứng được tách biệt về mặt logic với các tài nguyên khác. Phần cứng có thể là bất kỳ hệ thống nào như máy tính để bàn (có phần cứng và hệ điều hành riêng) được gọi là máy chủ, trên đó có thể chạy nhiều máy ảo hoặc máy khách, mỗi máy có hệ điều hành riêng. Điều này có thể thực hiện được nhờ một chương trình cơ sở được gọi là trình quản lý ảo.



Container giống như máy ảo, chạy trên máy chủ. Chúng cũng có thể được kiểm soát khi cần và cung cấp khả năng cô lập các ứng dụng khỏi môi trường thời gian chạy. Tuy nhiên, container hướng đến ảo hóa hệ điều hành. Không giống như máy ảo, trong hệ thống container, hệ điều hành khách không cần phải được cài đặt trên hệ thống máy chủ. Thay vào đó, hệ điều hành máy chủ cơ bản được chia sẻ bởi tất cả các phiên bản container chứa mã ứng dụng. Điều này làm cho container nhẹ hơn và giảm chi phí hệ điều hành liên quan đến máy ảo như vá lỗi, nâng cấp, hỗ trợ trình điều khiển, v.v. Hơn nữa, nó làm giảm lượng tài nguyên của máy chủ bị tiêu thụ bởi sự hiện diện của từng hệ điều hành.

**Sự khác biệt giữa Docker và máy ảo (VM) là gì?**

Sau đây là những khác biệt chính giữa [Docker](https://www.geeksforgeeks.org/introduction-to-docker/) và VM:

| **Diện mạo** | **Người lái tàu** | **Máy ảo (VM)** |
| --- | --- | --- |
| **Sự cách ly** | Docker đi kèm với nền tảng nhẹ,  sử dụng công nghệ container hóa cho các môi trường cô lập. | Máy ảo đi kèm phần mềm nặng cung cấp khả năng ảo hóa hoàn toàn để cô lập toàn bộ hệ điều hành. |
| **Sử dụng tài nguyên** | Nó tiêu thụ ít tài nguyên hơn từ hạt nhân Host OS và sử dụng chúng hiệu quả hơn | Máy ảo (VM) kém hiệu quả hơn trong việc sử dụng tài nguyên, mỗi VM sử dụng nhân hệ điều hành riêng và tiêu tốn nhiều tài nguyên hơn. |
| **Thời gian bắt đầu** | Những cách này nhanh hơn trong việc thiết lập nền tảng hệ điều hành (thùng chứa), thường chỉ mất vài giây. | Những giải pháp này khởi động chậm hơn và thường mất vài phút để thiết lập các Phiên bản. |
| **Khả năng mở rộng** | Việc mở rộng quy mô container trong Docker rất dễ dàng vì nó chiếm ít diện tích hơn. | Sẽ tốn nhiều tài nguyên khi mở rộng quy mô các phiên bản. |
| **Tính di động** | Docker cung cấp các ứng dụng được đóng gói có tính di động cao, nhất quán trên nhiều môi trường khác nhau. | Những hệ điều hành này cũng dễ di chuyển nhưng sẽ nặng hơn do phải bao gồm toàn bộ hệ điều hành. |
| **Trường hợp sử dụng** | Nó sẽ có hiệu quả đối với các dịch vụ siêu nhỏ, ứng dụng nhẹ và môi trường chứa container. | Thích hợp để chạy nhiều ứng dụng trong nhiều môi trường khác nhau. |

**Điểm tương đồng giữa Docker và VM**

Sau đây là những điểm tương đồng chính giữa Docker và Máy ảo:

* **Môi trường cô lập:** Cả Docker và Máy ảo đều cung cấp một số mức độ cô lập cho các ứng dụng. Docker đạt được sự cô lập thông qua việc chứa các quy trình cô lập trong khi VM đạt được sự ảo hóa hoàn toàn như hai hệ điều hành hoàn toàn riêng biệt.
* **Tính di động của ứng dụng:** Tính di động của ứng dụng có thể thực hiện được trên cả docker và máy ảo để chạy ứng dụng một cách nhất quán trên nhiều nền tảng khác nhau. Docker cung cấp [hình ảnh docker](https://www.geeksforgeeks.org/what-is-docker-image/) và VM cung cấp khả năng đóng gói toàn bộ [hệ điều hành ảo hóa](https://www.geeksforgeeks.org/virtual-memory-in-operating-system/) , cho phép di chuyển dễ dàng giữa nhiều nền tảng khác nhau.
* **Snapshot và Versioning:** Snapshot và Versioning được hỗ trợ trên cả hai nền tảng, cho phép người dùng nắm bắt trạng thái hiện tại của ứng dụng tại một thời điểm cụ thể. Tính năng này hữu ích để tạo bản sao lưu, thử nghiệm và tạo phiên bản, đồng thời cung cấp một cách hiệu quả để quay lại trạng thái đã biết cụ thể.
* **Quản lý tài nguyên:** Cả hai công nghệ đều cung cấp khả năng quản lý tài nguyên hiệu quả. Cả hai đều hỗ trợ phân bổ các tài nguyên cụ thể như CPU, bộ nhớ và lưu trữ, đảm bảo các ứng dụng chạy với các tài nguyên cần thiết mà không có bất kỳ sự can thiệp nào từ nhau.

**Lợi ích của Container**

* Các ứng dụng có thể được triển khai mà không cần lo lắng về môi trường thời gian chạy. Do đó, một ứng dụng có thể dễ dàng được di chuyển qua chu kỳ phát triển phần mềm và có thể chạy ở bất kỳ đâu, ví dụ, trên Mac OS, [Linux,](https://www.geeksforgeeks.org/introduction-to-linux-operating-system/) Windows và thậm chí trong các trung tâm dữ liệu. Điều này giúp giảm thời gian dành cho việc kiểm tra môi trường và có nhiều thời gian hơn cho việc phát triển chức năng mới.
* Nhiều container có ứng dụng có thể chạy trên cùng một phiên bản tài nguyên vật lý chia sẻ một hệ điều hành. Các container này nhẹ nên nhanh và sử dụng hiệu quả các tài nguyên điện toán có sẵn.
* Các container được tách biệt với nhau, cho phép nhà phát triển có thể chia các dịch vụ ứng dụng thành các container khác nhau. Các container này không chia sẻ bất kỳ sự phụ thuộc nào và mỗi container có thể được nhà phát triển tùy ý thao tác và cập nhật.
* Container có sẵn khả năng [kiểm soát phiên bản.](https://www.geeksforgeeks.org/version-control-systems/)

**Lợi ích của máy ảo**

* **Cô lập và bảo mật:** VM cung cấp khả năng cô lập mạnh mẽ bằng cách chạy các ứng dụng trên Hệ điều hành riêng biệt hoàn toàn ảo trên một máy vật lý duy nhất. Cô lập cung cấp khả năng bảo mật cho các ứng dụng trong môi trường.
* **Tính linh hoạt và khả năng tương thích:** VM hỗ trợ nhiều hệ điều hành khác nhau, cho phép nhiều ứng dụng chạy trên cùng một phần cứng vật lý. Tính linh hoạt của nó giúp nó phù hợp để lưu trữ nhiều ứng dụng theo môi trường được chỉ định.
* **Snapshots và Rollbacks:** Máy ảo hỗ trợ việc chụp snapshot và ghi lại toàn bộ trạng thái của VM tại một thời điểm cụ thể. Nó cung cấp cho người dùng khả năng khôi phục trạng thái đã biết và cần thiết của họ trong trường hợp cập nhật hoặc bất kỳ sự cố nào.
* **Tối ưu hóa tài nguyên:** Ảo hóa cung cấp khả năng phân bổ và sử dụng tài nguyên hiệu quả, cung cấp cấu hình động các tài nguyên phần cứng như [RAM,](https://www.geeksforgeeks.org/random-access-memory-ram/) CPU và Lưu trữ cho các lượng cụ thể. Tính linh hoạt của nó nâng cao khả năng quản lý tài nguyên và khả năng mở rộng trong môi trường ảo hóa.

**Khi nào nên sử dụng: Docker so với máy ảo**

**1. Sử dụng Docker trong các trường hợp sau**

* **Cô lập nhẹ là đủ:** Docker hiệu quả trong những tình huống mà cô lập quy trình nhẹ là đủ. Nó hiệu quả khi sử dụng với kiến ​​trúc microservices và cho [các ứng dụng container](https://www.geeksforgeeks.org/containerization-using-docker/) hiệu quả với chi phí tối thiểu.
* **Triển khai và mở rộng nhanh chóng:** Các container Docker sẽ khởi động các ứng dụng một cách nhanh chóng, phù hợp với các môi trường năng động đòi hỏi triển khai và mở rộng nhanh chóng.
* **Phát triển và triển khai nhất quán:** Docker cung cấp tính nhất quán giữa môi trường phát triển và triển khai cho các ứng dụng. Nó đảm bảo rằng các ứng dụng hoạt động theo cùng một cách trong các giai đoạn khác nhau của vòng đời phát triển.

**2. Sử dụng máy ảo trong các trường hợp sau**

* **Yêu cầu cô lập hoàn toàn hệ điều hành:** Nếu ứng dụng của bạn cần cô lập hoàn toàn như các hệ điều hành riêng biệt, thì Máy ảo là lựa chọn thích hợp. VM cung cấp khả năng cô lập mạnh hơn, giúp ứng dụng phù hợp để chạy trên các yêu cầu hệ điều hành khác nhau.
* **Ứng dụng cũ và khả năng tương thích: VM** được ưa chuộng để lưu trữ các loại ứng dụng cũ không thể dễ dàng chứa trong container. Chúng hỗ trợ nhiều Hệ điều hành khác nhau, giúp các ứng dụng tương thích với các phiên bản và cấu hình hệ điều hành cụ thể.
* **Khối lượng công việc đòi hỏi nhiều tài nguyên:** VM có thể được sử dụng trong các tình huống cần tính toán chuyên sâu cho khối lượng công việc, chúng ta có thể sử dụng VM để kiểm soát chi tiết việc phân bổ tài nguyên và làm cho chúng phù hợp hơn với các trường hợp.