Ex 1: Online Extraction vs Offline Extraction. Example?

**Online extraction:** Gathering data in real-time, suitable for monitoring and quick response, such as social media monitoring and real-time inventory management.

**Offline extraction:** Gathering data on a scheduled basis, suitable for long-term trend analysis and planning, such as monthly or yearly sales data analysis.

**Summary :** Trích xuất trực tuyến tập trung vào việc thu thập dữ liệu thời gian thực và phân tích ngay lập tức, trong khi trích xuất ngoại tuyến liên quan đến xử lý dữ liệu theo batch thu thập định kỳ để phân tích lịch sử và báo cáo. Sự lựa chọn giữa trích xuất trực tuyến và ngoại tuyến phụ thuộc vào các yếu tố như tính chất của nguồn dữ liệu, yêu cầu của phân tích và thời điểm mong muốn của thông tin.

**Example :**

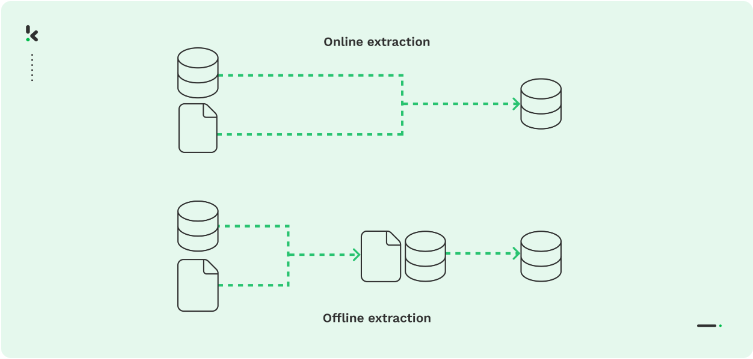
**Quản lý kho hàng của một nhà bán lẻ :**

1. **Trích xuất trực tuyến :**

* **Scenario:** Một nhà bán lẻ trực tuyến muốn duy trì thông tin kho hàng của họ theo thời gian thực để đảm bảo rằng họ có thể đáp ứng nhu cầu mua hàng của khách hàng ngay lập tức.
* **Phương Pháp:** Họ triển khai một hệ thống quản lý kho hàng kết nối trực tiếp với hệ thống đặt hàng và thanh toán của họ. Mỗi khi một đơn đặt hàng mới được đặt, hệ thống tự động cập nhật lượng hàng còn lại trong kho hàng.
* **Lợi Ích:** Nhờ vào trích xuất trực tuyến, nhà bán lẻ có thể theo dõi số lượng hàng còn lại trong kho hàng một cách chính xác và nhanh chóng, giúp họ điều chỉnh quy trình đặt hàng và lập kế hoạch vận chuyển một cách hiệu quả.

1. **Trích xuất ngoại tuyến :**

* **Scenario:** Một nhà bán lẻ lớn muốn phân tích xu hướng bán hàng trong năm qua để chuẩn bị kế hoạch cho mùa mua sắm sắp tới.
* **Phương Pháp:** Họ thiết lập một quy trình trích xuất dữ liệu hàng tháng từ hệ thống quản lý kho hàng của họ. Dữ liệu này bao gồm số lượng hàng nhập vào, bán ra, hàng tồn kho và doanh số bán hàng.
* **Lợi Ích:** Trích xuất ngoại tuyến cho phép nhà bán lẻ phân tích các xu hướng dài hạn trong bán hàng, nhận diện mô hình mua sắm của khách hàng, và tạo ra các chiến lược marketing và phân phối phù hợp cho tương lai.



*Source:* [*https://www.klippa.com/en/blog/information/dataextraction/*](https://www.klippa.com/en/blog/information/dataextraction/%23:~:text=Online%20extraction%3A%20There%20is%20a,outside%20of%20the%20source%20system.)

Ex2: Motivation of ELT model. What's the difference between ETL and ELT?

* **Hiệu suất và tính linh hoạt :** Trong quá trình xử lý dữ liệu truyền thống, dữ liệu thường được xử lý và biến đổi trước khi được tải vào kho dữ liệu. Mô hình ELT đảo ngược quá trình này bằng cách tải dữ liệu gốc vào kho dữ liệu trước, sau đó thực hiện các biến đổi và xử lý trên dữ liệu trong kho. Điều này thường mang lại hiệu suất cao và tính linh hoạt trong việc xử lý các loại dữ liệu đa dạng.
* **Giảm thời gian xử lý :** Bằng cách tải dữ liệu gốc trực tiếp vào kho dữ liệu mà không cần biến đổi trước, mô hình ELT giảm thời gian cần thiết để xử lý dữ liệu. Điều này giúp giảm bớt thời gian và công sức cho việc xử lý dữ liệu trước khi nó được sử dụng.
* **Hỗ trợ Big Data :** Với sự gia tăng về lượng dữ liệu được tạo ra mỗi ngày, việc sử dụng mô hình ELT giúp hỗ trợ hiệu quả hơn cho việc xử lý dữ liệu lớn. ELT cho phép các tổ chức xử lý và lưu trữ dữ liệu một cách hiệu quả hơn mà không cần phải giảm bớt hoặc tổ chức lại dữ liệu trước khi nó được tải vào kho dữ liệu.
* **Linh hoạt trong phân tích dữ liệu :** Bằng cách tải dữ liệu gốc vào kho dữ liệu trước, các nhà phân tích dữ liệu có thể truy cập vào dữ liệu gốc mà không bị ràng buộc bởi các biến đổi trước. Điều này mang lại tính linh hoạt cao trong việc thực hiện các phân tích và trích xuất thông tin từ dữ liệu.

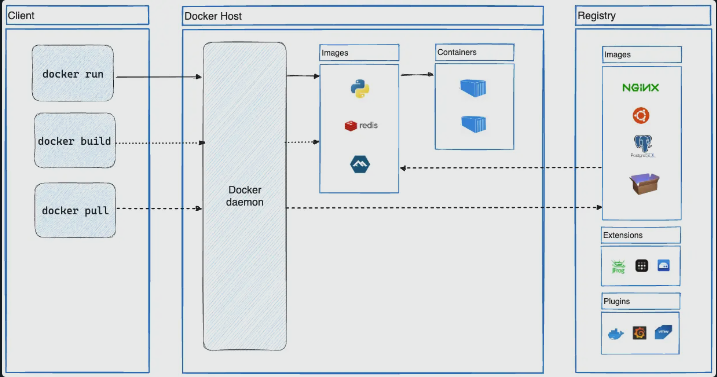
**Sự khác nhau giữa ETL và ELT ?**

|  |  |  |  |
| --- | --- | --- | --- |
| **Khác nhau** | **Thứ tự thực hiện công việc** | **Xử lý dữ liệu** | **Hiệu suất và quy mô** |
| **ETL** | Thực hiện theo thứ tự Extract, Transform, Load. Điều này có nghĩa là dữ liệu được trích xuất từ nguồn, sau đó được biến đổi (transform) theo các quy tắc và cuối cùng được tải vào kho dữ liệu. | Dữ liệu thường được biến đổi trước khi được tải vào kho dữ liệu. Các quy tắc xử lý dữ liệu có thể bao gồm việc chọn lọc, làm sạch, chuyển đổi định dạng, thống nhất dữ liệu, và tổ chức lại cấu trúc dữ liệu. | Thường được sử dụng cho các dự án có quy mô nhỏ đến trung bình, nơi mà dữ liệu có thể được biến đổi trước và có thể được lưu trữ ở dạng chuẩn trước khi tải vào kho dữ liệu. |
| **ELT** | Thực hiện theo thứ tự Extract, Load, Transform. Dữ liệu được trích xuất từ nguồn và sau đó được tải trực tiếp vào kho dữ liệu mà không cần biến đổi trước. Sau khi dữ liệu đã được tải vào kho, các biến đổi và xử lý được thực hiện trong kho dữ liệu. | Dữ liệu được tải trực tiếp vào kho dữ liệu mà không cần biến đổi trước. Các biến đổi và xử lý dữ liệu sau đó được thực hiện trong kho dữ liệu, thường sử dụng các công cụ và ngôn ngữ truy vấn như SQL hoặc công cụ xử lý dữ liệu phân tán. | Thường được sử dụng cho các dự án có quy mô lớn hoặc khi cần xử lý dữ liệu thô mà không cần biến đổi trước. ELT thường cho phép xử lý dữ liệu một cách phân tán và hiệu quả hơn đối với các kho dữ liệu lớn. |

Ex3: Investigate the overall architecture of Docker; The differences between Docker Container and Docker Image; How to build a docker image; How to run a container from a self-built docker image; How to run a container from docker hub image.

**Investigate the overall architecture of Docker :**

Docker uses a client-server architecture. The Docker client talks to the Docker daemon, which does the heavy lifting of building, running, and distributing your Docker containers. The Docker client and daemon can run on the same system, or you can connect a Docker client to a remote Docker daemon. The Docker client and daemon communicate using a REST API, over UNIX sockets or a network interface. Another Docker client is Docker Compose, that lets you work with applications consisting of a set of containers.



[https://docs.docker.com](https://docs.docker.com/get-started/overview/%23:~:text=Docker%20uses%20a%20client%2Dserver,to%20a%20remote%20Docker%20daemon.)

**The differences between Docker Container and Docker Image**

**Docker Image :**

Một Docker image là một gói thực thi độc lập và nhẹ mà chứa tất cả mọi thứ cần thiết để chạy một phần mềm, bao gồm mã nguồn, runtime, thư viện, biến môi trường và tập tin cấu hình. Nói cách khác, nó là một bản snapshot của một hệ thống tập tin bao gồm tất cả các thành phần cần thiết cho ứng dụng.

Một Docker image được xây dựng bằng cách sử dụng một tệp được gọi là Dockerfile, trong đó chỉ định các hướng dẫn cần thiết để tạo ra hình ảnh. Sau khi được xây dựng, image này có thể được lưu trữ trong một Docker registry như Docker Hub và từ đó có thể được kéo xuống để tạo ra các container trên bất kỳ máy chủ hỗ trợ Docker nào.

Image là không thể thay đổi và được coi là không thể sửa đổi một khi đã được tạo ra. Bất kỳ thay đổi nào được thực hiện trong một container, như sửa đổi tệp hoặc cài đặt phần mềm mới, sẽ không phản ánh trong hình ảnh gốc trừ khi tạo ra một image mới từ container đã được sửa đổi.

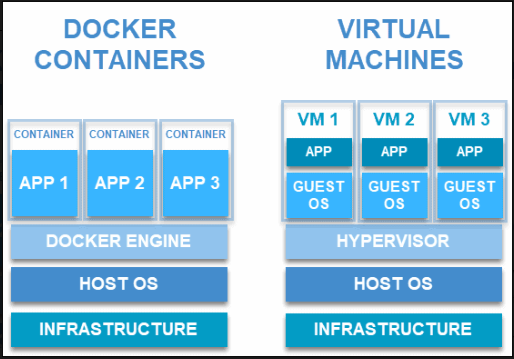
**Docker Container :**

Docker container là một run-time environment mà ở đó người dùng có thể chạy một ứng dụng độc lập. Những container này rất gọn nhẹ và cho phép bạn chạy ứng dụng trong đó rất nhanh chóng và dễ dàng.

Một tính năng quan trọng của container là tính chuẩn xác cho việc chạy các ứng dụng trong container. Không chỉ đảm bảo cho ứng dụng hoạt động như nhau trong các môi trường giống nhau, nó còn làm đơn giản việc cài đặt và chia sẻ cài đặt này cho các thành viên trong team.

Vì container hoạt động độc lập, nó đảm bảo không làm ảnh hưởng xấu đến các container khác, cũng như server mà nó đang chạy trong đó. Docker được cho là "tạo ra sự độc lập tuyệt vời". Vì vậy, bạn sẽ không cần lo lắng việc máy tính của bạn bị xung đột do ứng dụng đang được phát triển được chạy trong container.

Không giống như máy ao (VM) nơi mà sự ảo hóa (virtualization) xảy ra ở tầng phần cứng (hardware level), container chỉ ảo hóa ở lớp ứng dụng (app level). Nó có thể dùng 1 máy, chia sẻ kernel và giả môi trường để chạy process độc lập. Điều này làm cho container cực kì nhẹ, không chiếm nhiều tài nguyên của máy.



[https://viblo.asia/p/phan-biet-docker-image-vs-container](https://viblo.asia/p/phan-biet-docker-image-vs-container-bJzKmz4wZ9N%23_docker-image-la-gi-1)