

# Cloud for Data Storage and Management

TimeSeries Team

Ngày 17 tháng 8 năm 2025

## Mục lục

<b>I. Tổng quan Data Engineering</b>	<b>2</b>
1 Định nghĩa & vai trò	2
2 Khái niệm quan trọng	3
2.1 Data Engineering xử lý những loại dữ liệu gì? . . . . .	3
2.2 Data Engineering lưu trữ dữ liệu ở đâu? . . . . .	3
2.3 Data Pipeline là gì? . . . . .	4
2.4 Kỹ thuật ETL/ELT . . . . .	4
<b>II. Điện toán đám mây là gì?</b>	<b>6</b>
1 Định nghĩa & lợi ích	6
2 Có những loại dịch vụ điện toán đám mây nào?	7
2.1 IaaS . . . . .	7
2.2 PaaS . . . . .	7
2.3 SaaS . . . . .	7
<b>III. Command line (CLI) với Cloud</b>	<b>8</b>
1 Command line (CLI) là gì?	8
2 Làm thế nào tương tác với command line?	8
2.1 Một vài thao tác CLI . . . . .	9
3 CLI với AWS	10
3.1 Các thao tác CLI cơ bản với AWS . . . . .	10
3.2 AWS IAM là gì? . . . . .	11
3.3 Hướng dẫn thực hành . . . . .	11

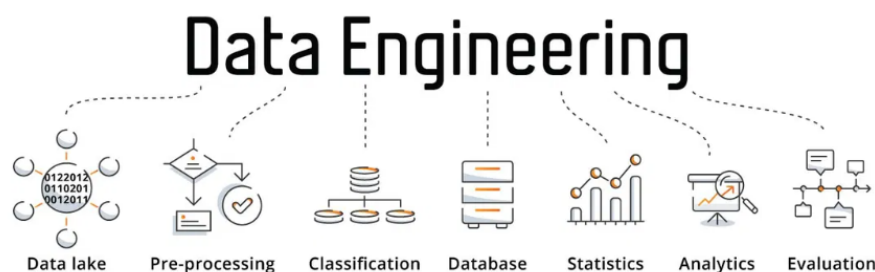
# I. Tổng quan Data Engineering

Data Engineering một trong những ngành nghề "hot" nhất hiện nay. Vậy thực chất làm Data Engineering là làm gì?

## 1 Định nghĩa & vai trò

**Data Engineering**, hay còn gọi là kỹ sư dữ liệu, là những người thiết kế, xây dựng và duy trì các hệ thống hạ tầng dữ liệu phức tạp. Các kỹ sư dữ liệu chịu trách nhiệm thu thập, chuyển đổi và lưu trữ dữ liệu từ nhiều nguồn khác nhau để tạo điều kiện thuận lợi cho các nhà phân tích dữ liệu (Data Analyst) và nhà khoa học dữ liệu (Data Scientist) thực hiện các phân tích chuyên sâu.

Cụ thể hơn, họ xây dựng những quy trình tự động để đảm bảo rằng dữ liệu được truyền tải và xử lý một cách chính xác và hiệu quả. Điều này đặc biệt quan trọng trong các tổ chức lớn với khối lượng dữ liệu khổng lồ và phức tạp.



### Vai trò chính của Data Engineer:

- Xây dựng và duy trì hệ thống dữ liệu: ETL và ELT: Dữ liệu thô thường không thể sử dụng ngay mà cần phải qua quá trình làm sạch và chuyển đổi thành dữ liệu có cấu trúc hoặc bán cấu trúc. Đây chính là nhiệm vụ của kỹ sư dữ liệu.
- Thiết kế và triển khai kho dữ liệu (Data Warehouse): Họ phát triển và duy trì các kiến trúc dữ liệu nhằm đảm bảo luồng thông tin từ các nguồn đầu vào tới các hệ thống lưu trữ và phân tích dữ liệu.
- Đảm bảo chất lượng và độ tin cậy của dữ liệu: đảm bảo rằng dữ liệu được bảo mật một cách tối đa, tránh rò rỉ hay bị tấn công từ các bên thứ ba.
- Tối ưu hóa hạ tầng dữ liệu: Kỹ sư dữ liệu luôn tìm cách cải tiến hiệu suất của hệ thống lưu trữ dữ liệu nhằm tăng tốc độ truy xuất và phân tích.

### Các kỹ năng cần thiết:

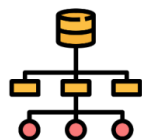
- Kỹ năng lập trình: Python, Java, Scala.
- Kiến thức về cơ sở dữ liệu: SQL và NoSQL.
- Kỹ năng xử lý dữ liệu lớn: Hadoop và Spark...
- Hiểu biết về hệ thống lưu trữ và phân phối: từ kho lưu trữ đám mây (AWS, Google Cloud) đến cơ sở dữ liệu nội bộ.

## 2 Khái niệm quan trọng

### 2.1 Data Engineering xử lý những loại dữ liệu gì?

Trong lĩnh vực Data Engineering, dữ liệu chính là “nguyên liệu thô” để xây dựng nên các hệ thống phân tích và trí tuệ nhân tạo. Tuy nhiên, không phải mọi dữ liệu đều giống nhau. Tùy theo cách tổ chức và định dạng, dữ liệu thường được chia thành ba nhóm chính:

1. Structured Data (có cấu trúc): Được tổ chức theo **schema** rõ ràng (hàng, cột), dễ dàng lưu trữ trong cơ sở dữ liệu quan hệ (RDBMS) hoặc Data Warehouse.
2. Unstructured Data (phi cấu trúc): Không có schema cố định, khó tổ chức theo hàng – cột.
3. Semi-structured Data (bán cấu trúc): Không cứng nhắc như Structured Data, nhưng có meta-data/tags giúp tổ chức. Thường thấy dưới dạng JSON hoặc XML.



Structured Data

Highly organized and follows a fixed format, like a spreadsheet or a relational database. It's easy to store, query, and analyze because it fits neatly into rows and columns with a defined schema



Unstructured Data

Has no predefined format or organization. It's the most common type of data today and includes things like text documents, emails, videos, images, and audio files



Semi-structured Data

Mix of both. It doesn't conform to a strict relational database model but contains tags or markers that organize the data and make it easier to process. Examples include JSON and XML files

Việc hiểu rõ đặc điểm của từng loại dữ liệu giúp Data Engineer lựa chọn hệ thống lưu trữ phù hợp, thiết kế data pipeline tối ưu, và đảm bảo dữ liệu sẵn sàng cho các nhu cầu phân tích, báo cáo, cũng như machine learning.

### 2.2 Data Engineering lưu trữ dữ liệu ở đâu?

Tất cả các kiểu dữ liệu phần trên sẽ được Data Engineer xử lý, vậy họ lưu dữ liệu ở đâu? Và cái kho này lưu thế nào?



A **data warehouse** is a central repository for **structured, processed data**. The data is cleaned, transformed, and organized into a specific schema before it is loaded. This is often referred to as a **schema-on-write** approach



A **data lake** is a centralized repository that stores all of an organization's data in its **native, raw format**, regardless of whether it's structured, semi-structured, or unstructured

Trong ảnh là sự so sánh giữa Data Warehouse và Data Lake – hai khái niệm quan trọng trong Data Engineering:

## 1. Data Warehouse:

- Là kho dữ liệu tập trung dành cho dữ liệu có cấu trúc (Structured Data), đã qua xử lý.
- Dữ liệu trước khi nạp vào kho sẽ được làm sạch, biến đổi và tổ chức theo một schema cố định (gọi là schema-on-write).
- Thích hợp cho báo cáo, phân tích BI (Business Intelligence), dashboard.

## 2. Data Lake:

- Là kho lưu trữ tập trung cho tất cả dữ liệu của tổ chức dưới dạng thô (raw format), bất kể là có cấu trúc, bán cấu trúc hay phi cấu trúc.
- Không cần định nghĩa schema trước khi lưu (schema-on-read).
- Thích hợp để lưu trữ dữ liệu lớn, đa dạng, phục vụ cho data science, machine learning, AI.

## 2.3 Data Pipeline là gì?

Việc kết hợp cơ sở dữ liệu, ngôn ngữ lập trình, công cụ xử lý và kho dữ liệu sẽ tạo ra một **pipeline**.

**Ví dụ:** Lấy dữ liệu bán hàng từ DB → xử lý bằng Spark → tính toán số liệu → đẩy kết quả vào Data Warehouse → đó chính là một pipeline.

Tuy nhiên, pipeline này không mấy hữu ích nếu bạn phải thực thi thủ công mỗi khi muốn nó chạy. Các data pipeline cần một trình cho phép chúng chạy auto vào các khoảng thời gian được chỉ định.

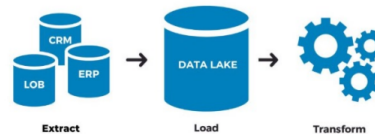
Data pipeline là một chuỗi các bước tự động dùng để di chuyển dữ liệu từ nguồn (source) đến điểm đích (destination).

## 2.4 Kỹ thuật ETL/ELT

Đường dẫn dữ liệu (**ETL** (Extract – Transform – Load) hay **ELT** (Extract – Load – Transform)) thực hiện nhiệm vụ tổng hợp, sắp xếp và di chuyển dữ liệu đến hệ thống mục tiêu nhằm tiến hành lưu trữ và phân tích.



**ETL (Extract, Transform, Load):** The traditional approach. Data is transformed on a separate server **before** being loaded into the data warehouse



**ELT (Extract, Load, Transform):** The modern, cloud-native approach. Raw data is loaded directly into a data lake, and transformation happens **within** the data warehouse. This leverages the scalability of the cloud.

Bất kể đó là ETL hay ELT, quá trình chuyển đổi/tích hợp dữ liệu bao gồm ba bước sau:

- Trích xuất (Extract): Lấy dữ liệu nguồn ra từ cơ sở dữ liệu gốc hoặc nguồn dữ liệu. Với ETL, dữ liệu đi vào vùng lưu trữ tạm thời. Với ELT, nó đi ngay vào hệ thống hồ dữ liệu (Data lake).
- Chuyển đổi (Transform): Thay đổi cấu trúc của thông tin. Quá trình chuyển đổi được tích hợp với hệ thống dữ liệu mục tiêu và phần còn lại của dữ liệu trong hệ thống đó.
- Truyền tải (Load): Gửi thông tin vào hệ thống lưu trữ dữ liệu.

### ETL là gì?

ETL là tên viết tắt của Extract, Transform và Load. Trong quá trình này, một công cụ ETL trích xuất dữ liệu từ các hệ thống nguồn RDBMS khác nhau sau đó chuyển đổi dữ liệu như áp dụng các biến đổi dữ liệu (tính toán, nối chuỗi v.v.) và sau đó tải dữ liệu vào hệ thống Data Warehouse. ETL là những luồng từ “nguồn” tới “đích”. Trong quá trình ETL, engine chuyển đổi sẽ xử lý mọi thay đổi dữ liệu.

### ELT là gì?

ETL là tên viết tắt của Extract, Load và Transform. Thay vì chuyển đổi dữ liệu trước khi tải, ELT cho phép “hệ thống đích” chuyển đổi trước. Dữ liệu đầu tiên được sao chép vào “đích” và sau đó được chuyển đổi tại đó. ELT thường được sử dụng với các database No-SQL như Hadoop, Data Appliance hoặc Cloud Installation.

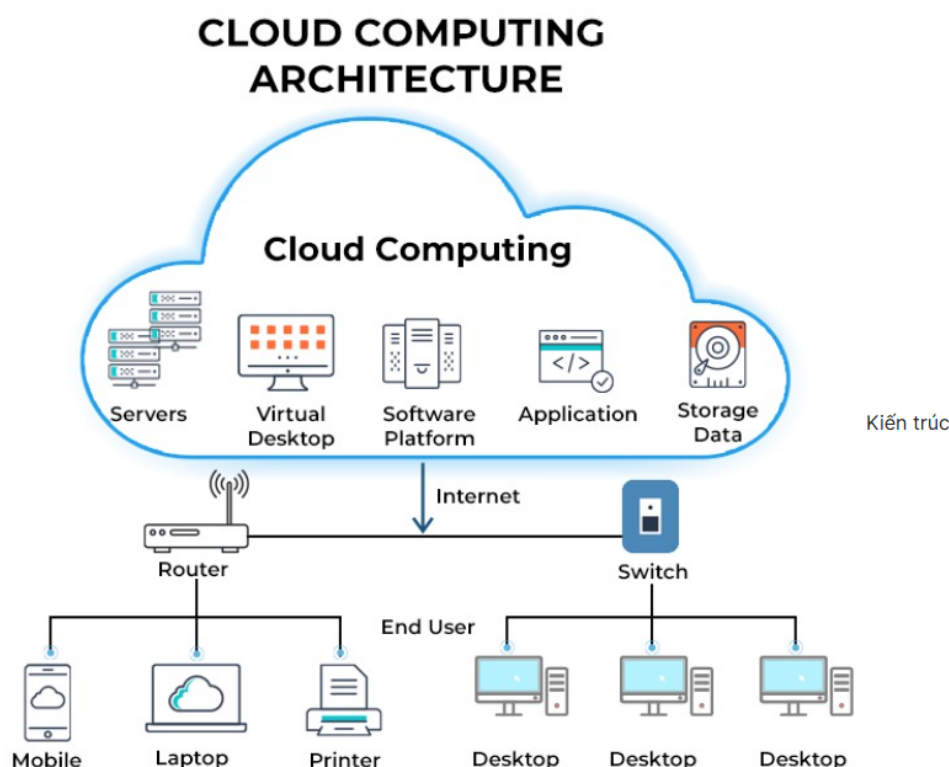
### Sự khác biệt giữa ETL và ELT

	ETL	ELT
1.Quy trình	Dữ liệu được chuyển đổi từ server staging sau đó được transfer tới Data warehouse DB	Dữ liệu vẫn còn trong DB của Data warehouse
2.Code Usage	Được sử dụng cho:-Những biến đổi chuyên sâu về tính toán- Lượng data nhỏ	Được sử dụng cho lượng data rất lớn
3.Biến đổi dữ liệu	Các biến đổi được thực hiện trong ETL server/staging	Các biến đổi được thực hiện bên trong “hệ thống đích”
4.Thời gian load	Dữ liệu trước tiên được load vào staging sau đó mới load vào “đích”. Cần nhiều thời gian	Dữ liệu được load vào “đích” chỉ 1 lần sau đó mới biến đổi. Nhanh hơn
5.Thời gian biến đổi	Quá trình ETL bắt buộc cần quá trình “Transform” hoàn tất. Khi kích thước dữ liệu tăng lên, thời gian chuyển đổi cũng tăng theo.	Trong quá trình ELT, tốc độ không bao giờ phụ thuộc vào kích thước của dữ liệu.
6.Thời gian bảo trì	Nhu cầu bảo trì là rất cao vì cần phải chọn dữ liệu để load và transform	Nhu cầu bảo trì là rất thấp vì dữ liệu luôn có sẵn
7.Độ phức tạp khi bắt đầu	Ở giai đoạn đầu thực hiện rất dễ dàng	Để thực hiện quá trình ELT, cần phải có những kiến thức rất sâu về các tools và kỹ năng chuyên môn
8.Hỗ trợ Data warehouse?	Mô hình ETL được sử dụng cho dữ liệu on-premise, quan hệ và có cấu trúc	Được sử dụng cho cơ sở hạ tầng cloud có thể support các nguồn dữ liệu có cấu trúc và phi cấu trúc
9.Hỗ trợ Data Lake	Không support	Cho phép sử dụng Data Lake với dữ liệu phi cấu trúc
10.Độ phức tạp	Quá trình ETL chỉ load những dữ liệu quan trọng, như đã được xác định trước từ thời điểm design	Quá trình này bao gồm tất cả quá trình phát triển từ output-backward và load những dữ liệu liên quan
11.Chỉ phí	Chỉ phí rất cao cho các doanh nghiệp vừa và nhỏ	Chỉ phí đầu vào thấp khi sử dụng các phần mềm online làm Services Platforms
12.Lookups	Trong quá trình ETL, cả 2 bảng Facts và Dimensions cần có sẵn trong Staging	Tất cả dữ liệu đều sẽ có sẵn vì Extract và Load được thực hiện chỉ trong 1 hành động
13. Aggregations	Độ phức tạp tăng lên với dữ liệu thêm vào trong dataset	Sức mạnh của target platform có thể xử lý một lượng dữ liệu đáng kể 1 cách nhanh chóng
14.Tính toán	Ghi đè lên cột đang có hoặc cần cấm cờ và đẩy sang “đích”	Dễ dàng thêm cột đã được tính toán vào bảng hiện có.
15.Maturity	ETL đã được sử dụng trong hơn 2 thập kỷ. Nó có bộ tài liệu tốt và dễ dàng để thực hành	Khái niệm tương đối mới và khá phức tạp để triển khai
16.Hardware	Hầu hết các tools đều có yêu cầu về hardware riêng biệt, tương đối đắt tiền	Chỉ phí cho phần cứng hệ thống điện toán đám mây không phải là vấn đề to tát
17. Hỗ trợ dữ liệu phi cấu trúc	Chủ yếu hỗ trợ dữ liệu quan hệ cấu trúc	Có hỗ trợ sẵn cho dữ liệu phi cấu trúc

## II. Điện toán đám mây là gì?

### 1 Định nghĩa & lợi ích

Về cơ bản, những công ty hay tập đoàn lớn thường cài đặt tất cả các ứng dụng và phần mềm thông qua 1 cụm máy chủ. 1 công ty sẽ có 1 hệ thống máy chủ, 2 công ty thì có 2 hệ thống, tương tự vậy với hàng trăm hay hàng ngàn công ty sẽ có lượng máy chủ tương ứng. Vấn đề đặt ra là nếu một công ty có hàng ngàn trụ sở trên khắp thế giới thì số lượng hệ thống máy chủ là rất lớn. Để giảm tải các chi phí phát sinh từ việc xây dựng các hệ thống đó, điện toán đám mây đã ra đời. “Đám mây” ở đây chính là cả hệ thống Internet – 1 mạng lưới khổng lồ vô tận.



**Điện toán đám mây** (tiếng Anh: cloud computing), còn gọi là điện toán máy chủ ảo, là mô hình điện toán sử dụng công nghệ máy tính và phát triển dựa vào mạng Internet.

Hiện nay hầu hết các doanh nghiệp đều sử dụng điện toán đám mây để đăng tải các ứng dụng làm việc của họ lên Internet, đồng thời thêm nhiều tính năng mới thông qua trình duyệt web.

#### Lợi ích của điện toán đám mây là gì?

- **Tối ưu hóa chi phí:** Chuyển sang sử dụng điện toán đám mây mang lại lợi ích to lớn cho các doanh nghiệp trong việc tối ưu hóa chi phí CNTT. Lý do chính nằm ở khả năng loại bỏ các khoản chi phí vốn khổng lồ liên quan đến việc mua sắm và vận hành cơ sở hạ tầng CNTT truyền thống.
- **Tốc độ cao:** Tốc độ triển khai dịch vụ vô cùng nhanh chóng và linh hoạt. Hầu hết các dịch vụ đều được cung cấp theo mô hình tự phục vụ và theo yêu cầu, cho phép doanh nghiệp dễ dàng truy cập và sử dụng tài nguyên máy tính một cách nhanh chóng và hiệu quả.

- Quy mô toàn cầu: Mở rộng quy mô hoạt động một cách linh hoạt và hiệu quả.
- Độ tin cậy cao: Điện toán đám mây mang đến giải pháp bảo vệ dữ liệu và đảm bảo tính liên tục trong kinh doanh hiệu quả hơn, giúp doanh nghiệp an tâm hoạt động và phát triển.
- Bảo mật toàn diện: Điện toán đám mây mang đến nhiều giải pháp bảo mật tiên tiến, giúp doanh nghiệp bảo vệ dữ liệu, ứng dụng và cơ sở hạ tầng một cách hiệu quả.

## 2 Có những loại dịch vụ điện toán đám mây nào?

Điện toán đám mây có thể được phân thành 3 loại chính:

1. Cơ sở hạ tầng dưới dạng dịch vụ – IaaS (Infrastructure as a Service)
2. Nền tảng dưới dạng dịch vụ – PaaS (Platform as a Service)
3. Phần mềm dưới dạng dịch vụ – SaaS (Software as a Service)

### 2.1 IaaS

Đóng vai trò quan trọng trong hệ thống Điện toán Đám mây, cung cấp những khối xây dựng cơ bản cho Công nghệ Thông tin và Truyền thông (CNTT). IaaS mở ra quyền truy cập vào các tính năng mạng, máy tính (ảo hoặc trên phần cứng chuyên dụng), và không gian lưu trữ dữ liệu.

IaaS nổi bật với tính linh hoạt cao và khả năng kiểm soát tài nguyên CNTT. Người dùng có khả năng quản lý và điều chỉnh tài nguyên theo nhu cầu cụ thể mà không gặp phải các ràng buộc về phần cứng. Mô hình này giống với cách nhiều tổ chức và nhà phát triển quản lý tài nguyên CNTT trong môi trường hiện tại, tạo điều kiện thuận lợi cho sự chuyển đổi và tích hợp vào mô hình đám mây. Điều này làm cho IaaS trở thành một phần không thể thiếu trong hệ thống Điện toán Đám mây, đặc biệt đối với những người muốn giữ sự linh hoạt và kiểm soát tối đa trong quản lý tài nguyên CNTT của họ.

### 2.2 PaaS

**PaaS** giúp giảm bớt gánh nặng quản lý cơ sở hạ tầng cơ bản của tổ chức, thường là phần cứng và hệ điều hành. Thay vào đó, nó tập trung vào việc triển khai và quản lý ứng dụng.

Với PaaS, người dùng có thể tập trung vào công việc chính như phát triển ứng dụng mà không cần lo lắng về việc mua sắm tài nguyên, quản lý dung lượng, bảo trì phần mềm và xử lý sự cố. PaaS giúp tối ưu hóa quá trình làm việc bằng cách giảm bớt các công đoạn quản lý cấp thấp, đồng thời mang lại sự thuận tiện và linh hoạt trong việc phát triển ứng dụng.

### 2.3 SaaS

**SaaS** cung cấp cho người dùng quyền truy cập vào các ứng dụng phần mềm sẵn sàng sử dụng. Người dùng có thể truy cập các ứng dụng này từ bất kỳ đâu có kết nối Internet.

SaaS thường được sử dụng bởi các doanh nghiệp và cá nhân. Ví dụ, các doanh nghiệp có thể sử dụng SaaS để triển khai các ứng dụng như email, CRM hoặc ERP. Các cá nhân có thể sử dụng SaaS để truy cập các ứng dụng như email, lưu trữ đám mây hoặc các dịch vụ phát trực tuyến.

Ngoài ra, điện toán đám mây còn có thể được phân loại theo mô hình triển khai:

- Public cloud
- Private cloud
- Hybrid cloud
- Community cloud



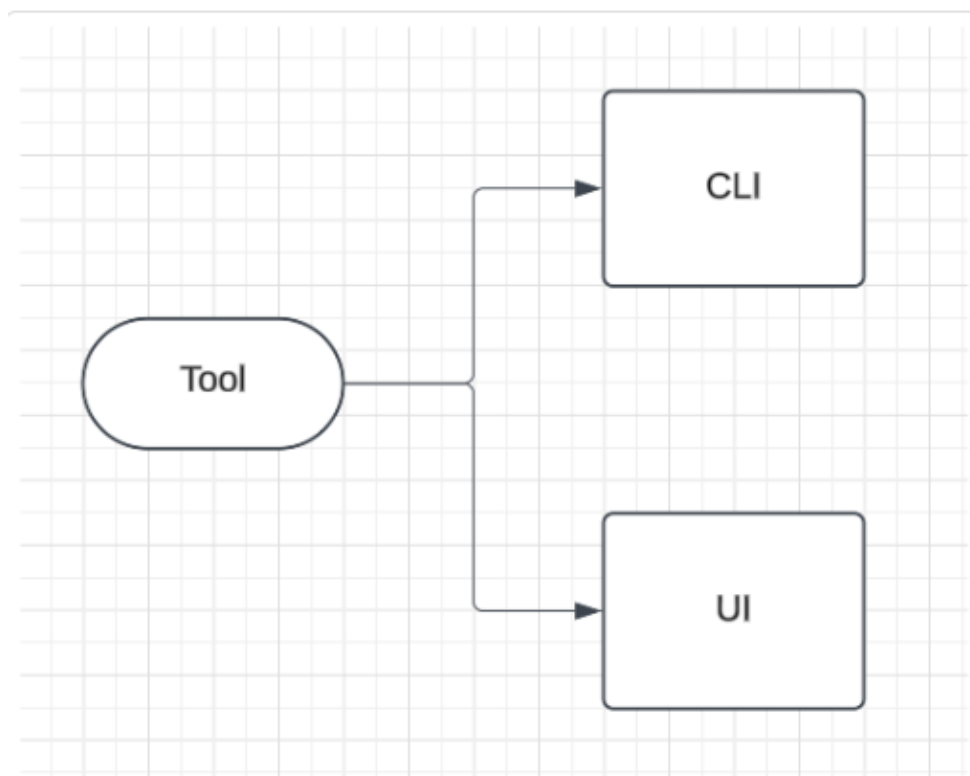
### III. Command line (CLI) với Cloud

#### 1 Command line (CLI) là gì?

- **Command** là lệnh.
- **Command line** là dòng lệnh.

Trước đây, khi máy tính còn là những cỗ máy to kềnh, không màu mè thì con người chủ yếu sử dụng dòng lệnh để giao tiếp với máy tính. Nhưng khi Windows ra đời thì nó nhanh chóng phổ cập hệ điều hành với giao diện đồ họa -> làm cho người dùng mất luôn khả năng giao tiếp bằng command line và không biết đến sự tồn tại của nó nữa.

Một công cụ phần mềm (software tool) thì thường có 2 kiểu: CLI và UI. Cũng có tool chỉ có UI, và ngược lại.



CLI thường được sử dụng trong các trường hợp configuration bằng code, các task lặp đi lặp lại. Ví dụ dự án mình làm là desktop app, khi có bản build mới mình phải uninstall bản cũ rồi install bản mới để test. Nếu mình cứ làm qua UI thì rất mất thời gian (đi qua rất nhiều form và dialog), thay vào đó mình chỉ run 1 câu lệnh là đã uninstall và 1 câu lệnh nữa là đã install được bản mới.

#### 2 Làm thế nào tương tác với command line?

Nếu bạn có 1 tool mà nó support command line, nhưng làm thế nào để tương tác với các tool đó, bạn cần có **Shell**. Shell là tool đứng giữa bạn và hệ điều hành (OS), nó sẽ gọi đến OS và từ OS gọi đến CLI tool của bạn. **User -> Shell -> OS -> CLI tool**

- Shell ở trong Windows có Command Prompt (CMD) hoặc Powershell

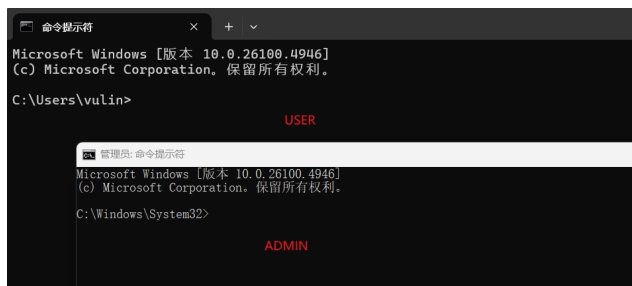


- Ở Linux thì có bash (default), Kornshell...
- Ở MacOS thì có bash, zsh (default)...

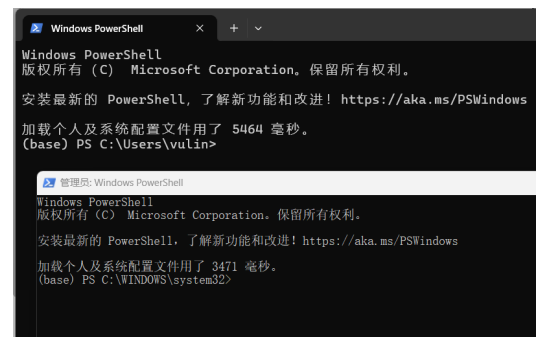
## 2.1 Một vài thao tác CLI

Vì mình dùng Windows nên mình sẽ thử vài lệnh của CMD và Powershell nhé. Mở khung search và gõ cmd/Powershell, có 2 chế độ mở:

- Open: mở mà vị trí đứng sẽ ở folder User
- Run as administrator: mở mà vị trí đứng sẽ ở folder System.



Hình 1: CMD



Hình 2: Powershell

Một số dòng lệnh cơ bản như:

- pwd: Cho ta thấy mình đang ở đâu trên máy tính/ current paths
- cd : thay đổi vị trí đứng.
- ls: hiển thị list tập tin tại chỗ đang đứng.

```
(base) PS C:\Users\vulin> pwd
Path
----
C:\Users\vulin

(base) PS C:\Users\vulin> cd d:
(base) PS D:\> pwd
Path
----
D:\

(base) PS D:\> cd c:
(base) PS C:\Users\vulin> ls

    目录: C:\Users\vulin

Mode                LastWriteTime         Length Name
----                -
d-----          2025/3/3   14:17          .anaconda
d-----          2025/2/25   22:18          .android
d-----          2025/3/5    14:38          .astrophy
d-----          2025/5/26   18:27          .bundle
d-----          2025/7/14   14:29          .cache
```

- mkdir <name>: khởi tạo thư mục mới.

- `rmdir <name>`: xóa tập tin.
- `cd ..`: quay về một bước, quay về vị trí thư mục trước đó
- `cp <file_name> <folder_name>`: copy 1 file sang 1 folder khác
- `mv <file1> <file2>`: chuyển file 1 sang chỗ file 2.
- `rm <filename>`: xóa filename.
- Command line - đọc file:
  - `less <filename>`: hiển thị page 1 của tài liệu.
  - `cat <filename>`: hiển thị toàn bộ.
  - `grep <word>`: tìm từ.
- history: hiển thị toàn bộ lịch sử command line.
- và còn rất nhiều nữa các bạn có thể tra thêm nhé.

Function	Command	Description
Other	> history	Show all command history
Other	> clear	clear the content display in terminal
Other	> whoami	Show the name of current user
Other	> cal	Show calendar
Other	> shutdown	Shutdown a computer
Other	> reboot	Restart a computer
Other	> htop	Monitor process viewer and process manager

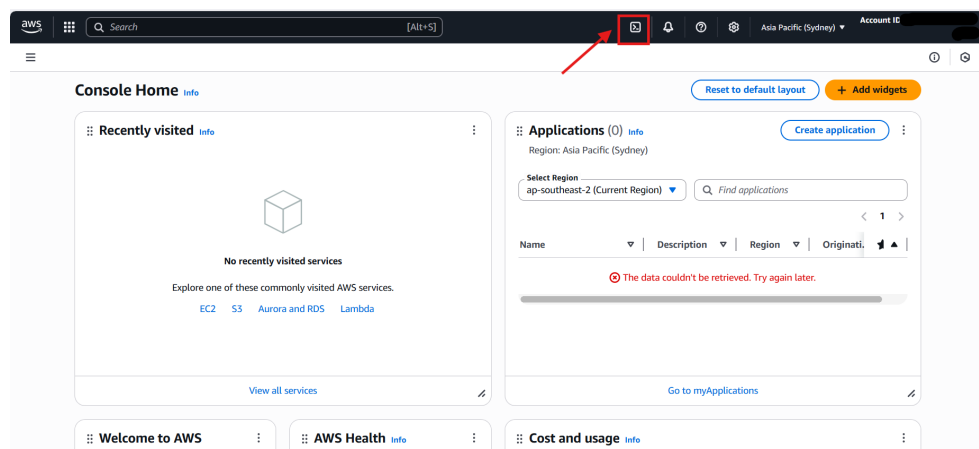
### ❖VIM editor

Command	Description
i	Insert mode (start typing text)
Esc	Back to Normal mode
v	Visual mode (select text)
:w	Save file
:q	Quit Vim
:wq	Save and quit
:q!	Quit without saving

## 3 CLI với AWS

### 3.1 Các thao tác CLI cơ bản với AWS

Trước khi thực hành thì các bạn hãy tạo một tài khoản AWS cho mình qua link: [AWS](#)  
 Sau khi tạo tài khoản xong bạn sẽ thấy giao diện AWS như sau:



Hãy nhấn vào biểu tượng mũi tên chỉ vào và thao tác CLI với AWS thôi nào ! Có rất nhiều câu lệnh các bạn có thể đọc tại [AWS CLI Command](#)



```
CloudShell
ap-southeast-2 +
~ $ pwd
/home/cloudshell-user
~ $ ls
~ $ mkdir test
~ $ ls
test
~ $ cd test
test $ pwd
/home/cloudshell-user/test
test $
```

Tương tự với các cloud khác nhau có các câu lệnh khác nhau, các bạn nhớ đọc documentation để hiểu kỹ hơn nhé.

### 3.2 AWS IAM là gì?

**AWS IAM** hay **AWS Identity and Access Management** chính là một dịch vụ web giúp doanh nghiệp có thể quản lý tập trung và kiểm soát chặt chẽ được quyền truy cập vào các tài nguyên Amazon Web Services một cách an toàn. Triển khai IAM AWS, doanh nghiệp có thể kiểm soát đầu là đối tượng được xác thực để đăng nhập tài khoản và được ủy quyền sử dụng các tài nguyên của chúng ta.

Trong AWS IAM cần lưu ý một số khái niệm sau:

- **IAM User:** “nick” dài hạn cho người/app (có password/access key). Dùng cho người là chính; app nên dùng Role.
- **IAM Group:** nhóm các User. Gắn policy vào Group để áp cho cả nhóm.
- **IAM Role:** danh tính tạm thời, không có key dài hạn. Được assume bởi user/dịch vụ (EC2, Lambda, cross-account...). Best practice cho app.
- **IAM Policy:** file JSON mô tả quyền (Allow/Deny). Gắn vào User/Group/Role (identity-based) hoặc gắn thẳng vào tài nguyên (resource-based, ví dụ S3 bucket policy).

### 3.3 Hướng dẫn thực hành

Để thực hành, các bạn có thể truy cập [Tìm các hướng dẫn thực hành cho nhu cầu về AWS của bạn](#)

Tại đây các bạn có thể tìm thấy mọi hướng dẫn chi tiết về quản lý tài khoản, cơ sở dữ liệu,...

The screenshot shows the AWS Learning Path page. At the top, there's a navigation bar with the AWS logo and links like 'Sản phẩm', 'Giải pháp', 'Định giá', 'Tài liệu', 'Tìm hiểu', 'Mạng lưới đối tác', 'AWS Marketplace', 'Tiếp cận khách hàng', 'Sự kiện', 'Khám phá thêm'. Below this is a search bar and a 'Đăng nhập vào hàng đầu tiên' button. The main heading is 'Tìm các hướng dẫn thực hành cho nhu cầu về AWS của bạn' (Find hands-on guides for your AWS needs). Below this, there's a section 'Bắt đầu với trung tâm tài nguyên' (Get started with the resource center) and a link 'Trung tâm nhà phát triển / Bắt đầu / ...'. The main content area is divided into two columns. The left column has a 'Lọc theo' (Filter by) section with a 'Danh mục' (Category) dropdown and a 'Cấp độ độ phức tạp' (Complexity level) dropdown. The right column shows a grid of tutorial cards. Each card has a title, a 'Thực hành' (Hands-on) button, a description, and a duration. The cards are: 'LƯU TRỮ trang web tĩnh' (Static website storage), 'BIỆN TOÁN' (Scripting), 'DI ĐỘNG' (Mobile), 'LƯU trữ và truy xuất tệp' (Storage and retrieval), and 'SERVERLESS' (Serverless). Each card also has a 'Xóa tất cả' (Clear all) button.