**Nguyễn Hoàng Vỹ - 22110275**

Module 6 : **Compute**

**1. Khái niệm**

* **Compute là gì?**
  + Đề cập đến khả năng xử lý, bộ nhớ và tài nguyên mạng cần thiết để chạy ứng dụng.
  + AWS cung cấp nhiều mô hình dịch vụ điện toán khác nhau để phù hợp với các nhu cầu đa dạng.
* **Amazon EC2 (Elastic Compute Cloud):**
  + **Khái niệm:** Dịch vụ cung cấp các máy chủ ảo có thể thay đổi kích thước (gọi là *instances*) trên đám mây. Đây là nền tảng cho nhiều dịch vụ AWS khác.
  + **Instance Types:** Các cấu hình khác nhau của CPU, bộ nhớ, lưu trữ và khả năng mạng (ví dụ: General Purpose, Compute Optimized, Memory Optimized, Storage Optimized, Accelerated Computing). Lựa chọn loại instance phù hợp với yêu cầu của ứng dụng.
  + **Amazon Machine Images (AMIs):** Khuôn mẫu (template) chứa hệ điều hành và phần mềm cần thiết để khởi chạy một instance. Có thể sử dụng AMIs do AWS cung cấp, từ Marketplace hoặc tự tạo.
  + **Storage Options for EC2:**
    - **Elastic Block Store (EBS):** Lưu trữ dạng khối (block storage) bền bỉ, hiệu năng cao, giống như ổ cứng mạng, tồn tại độc lập với instance. Có nhiều loại EBS khác nhau (gp2/gp3, io1/io2, st1, sc1).
    - **Instance Store:** Lưu trữ tạm thời (ephemeral), nằm trên cùng máy chủ vật lý với instance. Dữ liệu sẽ mất khi instance dừng hoặc chấm dứt. Tốc độ truy cập rất nhanh.
  + **Security Groups:** Tường lửa ảo (virtual firewall) ở cấp độ instance, kiểm soát lưu lượng truy cập vào và ra (inbound/outbound traffic) dựa trên các quy tắc (rules) về cổng (port) và địa chỉ IP.
  + **Key Pairs:** Sử dụng để xác thực khi bạn kết nối vào instance EC2 (ví dụ: qua SSH cho Linux hoặc RDP cho Windows). Bao gồm một public key (lưu trên AWS) và một private key (bạn tự lưu trữ an toàn).
  + **EC2 Pricing Models:**
    - **On-Demand:** Trả tiền theo giờ hoặc giây sử dụng, không có cam kết dài hạn. Linh hoạt nhất.
    - **Reserved Instances (RIs):** Cam kết sử dụng trong 1 hoặc 3 năm để đổi lấy mức giảm giá đáng kể so với On-Demand.
    - **Savings Plans:** Mô hình giá linh hoạt hơn RI, cam kết chi tiêu một lượng tiền nhất định (tính bằng $/giờ) trong 1 hoặc 3 năm để được giảm giá trên EC2, Fargate và Lambda.
    - **Spot Instances:** Tận dụng dung lượng EC2 chưa sử dụng với mức giảm giá rất lớn (lên đến 90%). Tuy nhiên, AWS có thể lấy lại instance này chỉ với 2 phút thông báo trước. Phù hợp cho các workload linh hoạt, có thể bị gián đoạn (ví dụ: batch processing, data analysis).
* **Container Services:**
  + **Containers:** Đóng gói ứng dụng và các thành phần phụ thuộc của nó lại với nhau, đảm bảo chạy nhất quán trên các môi trường khác nhau. (Docker là công nghệ container phổ biến).
  + **Amazon ECS (Elastic Container Service):** Dịch vụ điều phối container (container orchestration) được quản lý hoàn toàn của AWS. Giúp dễ dàng chạy, dừng và quản lý các Docker container trên một cụm (cluster) EC2 instances hoặc sử dụng AWS Fargate.
  + **Amazon EKS (Elastic Kubernetes Service):** Dịch vụ Kubernetes được quản lý của AWS. Cho phép chạy Kubernetes trên AWS mà không cần tự cài đặt và quản lý control plane.
  + **AWS Fargate:** Công nghệ điện toán serverless cho container (hoạt động với cả ECS và EKS). Cho phép chạy container mà không cần quản lý các EC2 instance bên dưới. Bạn chỉ cần định nghĩa container và tài nguyên cần thiết.
* **Serverless Computing:**
  + **AWS Lambda:** Dịch vụ điện toán serverless cho phép chạy mã nguồn (code) mà không cần cấp phát hoặc quản lý máy chủ. Bạn chỉ trả tiền cho thời gian tính toán thực tế mà mã nguồn của bạn chạy. Lambda tự động mở rộng quy mô dựa trên yêu cầu. Được kích hoạt bởi các sự kiện (ví dụ: thay đổi dữ liệu trong S3, yêu cầu HTTP qua API Gateway).
* **Platform as a Service (PaaS):**
  + **AWS Elastic Beanstalk:** Dịch vụ dễ sử dụng để triển khai và mở rộng quy mô các ứng dụng web và dịch vụ được phát triển bằng Java, .NET, PHP, Node.js, Python, Ruby, Go và Docker trên các máy chủ quen thuộc như Apache, Nginx, Passenger và IIS. Bạn chỉ cần tải lên mã nguồn của mình và Elastic Beanstalk tự động xử lý việc triển khai, từ cấp phát tài nguyên, cân bằng tải, tự động mở rộng quy mô đến giám sát tình trạng ứng dụng.

**2. Tóm tắt nội dung video**

* **Introduction Video:** Giới thiệu về tầm quan trọng của điện toán trong cloud và các chủ đề chính sẽ được đề cập trong module.
* **Section 1 Video - Compute Services Overview:** Cung cấp cái nhìn tổng quan về các loại dịch vụ điện toán khác nhau trên AWS (EC2, Containers, Lambda, Elastic Beanstalk) và khi nào nên sử dụng từng loại.
* **Section 2, 3, 4 Video - Amazon EC2 Part 1, 2, 3:** Đi sâu vào dịch vụ EC2:
  + Part 1: Giải thích EC2 là gì, các loại instance (Instance Types) và trường hợp sử dụng của chúng.
  + Part 2: Trình bày về AMIs (cách chọn và sử dụng) và các lựa chọn lưu trữ đi kèm EC2 (EBS vs Instance Store).
  + Part 3: Tập trung vào bảo mật (Security Groups, Key Pairs) và các khía cạnh mạng cơ bản liên quan đến EC2 instances.
* **Console Demonstration - EC2:** Hướng dẫn trực quan các bước để khởi chạy, kết nối và quản lý một EC2 instance thông qua AWS Management Console.
* **Section 5 Video - Amazon EC2 Cost Optimization:** Thảo luận về các mô hình định giá khác nhau của EC2 (On-Demand, RIs, Savings Plans, Spot) và cách chọn mô hình phù hợp để tiết kiệm chi phí.
* **Section 6 Video - Container Services:** Giới thiệu khái niệm container và các dịch vụ điều phối container của AWS như ECS, EKS và Fargate, giải thích lợi ích và sự khác biệt giữa chúng.
* **Section 7 Video - Introduction to AWS Lambda:** Giải thích khái niệm điện toán serverless và cách Lambda hoạt động, các trường hợp sử dụng phổ biến (xử lý sự kiện, backend cho API).
* **Section 8 Video - Introduction to AWS Elastic Beanstalk:** Trình bày về Elastic Beanstalk như một giải pháp PaaS giúp đơn giản hóa việc triển khai và quản lý ứng dụng web, giải thích cách nó tự động hóa các tác vụ vận hành.
* **Wrap Up Video:** Tóm tắt lại các dịch vụ và khái niệm điện toán chính đã học trong module.

**3. Các Case Study (Ví dụ thực tế)**

* **Case Study 1: Hosting Website/Ứng dụng Web (Sử dụng EC2)**
  + *Tình huống:* Một công ty cần host website công ty hoặc một ứng dụng web tùy chỉnh. Họ cần toàn quyền kiểm soát hệ điều hành và môi trường chạy.
  + *Giải pháp:* Sử dụng Amazon EC2 để khởi chạy một hoặc nhiều instance (ví dụ: Linux với Apache/Nginx hoặc Windows với IIS). Cấu hình Security Groups để cho phép truy cập HTTP/HTTPS. Sử dụng EBS cho lưu trữ bền bỉ. Có thể kết hợp với Elastic Load Balancing và Auto Scaling để tăng tính sẵn sàng và khả năng mở rộng.
  + *Lợi ích:* Toàn quyền kiểm soát môi trường, linh hoạt trong cấu hình, khả năng mở rộng theo nhu cầu.
* **Case Study 2: Xử lý Dữ liệu Hàng loạt (Sử dụng EC2 Spot Instances)**
  + *Tình huống:* Một công ty nghiên cứu cần xử lý một lượng lớn dữ liệu基因组 (genome data) hoặc thực hiện các mô phỏng phức tạp. Công việc này không yêu cầu phải hoàn thành ngay lập tức và có thể bị gián đoạn.
  + *Giải pháp:* Sử dụng EC2 Spot Instances để tận dụng giá rẻ. Thiết kế quy trình xử lý sao cho có thể lưu lại trạng thái và tiếp tục khi instance mới được cấp phát nếu instance hiện tại bị thu hồi.
  + *Lợi ích:* Giảm chi phí tính toán đáng kể cho các workload linh hoạt.
* **Case Study 3: Triển khai Microservices (Sử dụng ECS/EKS + Fargate)**
  + *Tình huống:* Một công ty công nghệ đang xây dựng ứng dụng theo kiến trúc microservices. Họ muốn đóng gói từng dịch vụ vào container để dễ quản lý, triển khai và mở rộng quy mô độc lập.
  + *Giải pháp:* Sử dụng Docker để tạo container images. Triển khai và quản lý các container này bằng Amazon ECS hoặc EKS. Sử dụng AWS Fargate để không cần phải quản lý các EC2 instance bên dưới, tập trung hoàn toàn vào ứng dụng.
  + *Lợi ích:* Quản lý container hiệu quả, dễ dàng mở rộng quy mô từng dịch vụ, giảm gánh nặng vận hành (đặc biệt với Fargate).
* **Case Study 4: Xây dựng Backend API Serverless (Sử dụng Lambda + API Gateway)**
  + *Tình huống:* Một ứng dụng di động cần một backend API để xử lý yêu cầu từ người dùng (ví dụ: đăng nhập, lấy dữ liệu). Lưu lượng truy cập có thể biến động mạnh. Công ty muốn tối ưu chi phí và không muốn quản lý máy chủ.
  + *Giải pháp:* Xây dựng các API endpoints bằng Amazon API Gateway. Mỗi endpoint sẽ kích hoạt một hàm AWS Lambda tương ứng để xử lý logic nghiệp vụ.
  + *Lợi ích:* Chi phí thấp (chỉ trả khi API được gọi và Lambda chạy), tự động mở rộng quy mô vô hạn, không cần quản lý hạ tầng server.
* **Case Study 5: Triển khai nhanh Ứng dụng Web Đơn giản (Sử dụng Elastic Beanstalk)**
  + *Tình huống:* Một nhóm phát triển nhỏ cần nhanh chóng triển khai một ứng dụng web PHP hoặc Node.js lên môi trường production mà không muốn tốn nhiều thời gian cấu hình hạ tầng (load balancer, auto scaling, monitoring).
  + *Giải pháp:* Đóng gói mã nguồn ứng dụng và tải lên AWS Elastic Beanstalk. Elastic Beanstalk sẽ tự động cấp phát và cấu hình các tài nguyên AWS cần thiết (EC2, ELB, Auto Scaling Group, RDS - nếu cần) và triển khai ứng dụng.
  + *Lợi ích:* Triển khai nhanh chóng, đơn giản hóa quản lý, tự động hóa nhiều tác vụ vận hành, giúp nhà phát triển tập trung vào code.

Module 7 : **Storage**

**1. Khái niệm :**

Module này tập trung vào các giải pháp lưu trữ khác nhau mà AWS cung cấp, đáp ứng các nhu cầu đa dạng về hiệu năng, độ bền, khả năng truy cập và chi phí. Các khái niệm chính bao gồm:

* **Các loại Lưu trữ trên Cloud:**
  + **Block Storage (Lưu trữ khối):** Cung cấp dữ liệu dưới dạng các khối (blocks), tương tự như ổ cứng vật lý (HDD/SSD) hoặc mạng lưu trữ SAN. Thường được gắn vào một máy chủ cụ thể. Phù hợp cho hệ điều hành, cơ sở dữ liệu, hệ thống file cần hiệu năng cao và độ trễ thấp.
  + **Object Storage (Lưu trữ đối tượng):** Lưu trữ dữ liệu dưới dạng các đối tượng (objects), bao gồm dữ liệu gốc, metadata, và một mã định danh duy nhất. Có khả năng mở rộng gần như vô hạn, độ bền cao, truy cập qua API (thường là HTTP/S). Phù hợp cho lưu trữ file lớn, backup, data lakes, static web hosting.
  + **File Storage (Lưu trữ file):** Cung cấp một hệ thống file chia sẻ có thể được truy cập bởi nhiều máy chủ cùng lúc thông qua các giao thức mạng chuẩn (như NFS hoặc SMB). Tương tự như Network Attached Storage (NAS). Phù hợp cho chia sẻ dữ liệu, ứng dụng cần hệ thống file dùng chung.
* **Các Dịch vụ Lưu trữ chính trên AWS:**
  + **Amazon EBS (Elastic Block Store):**
    - **Khái niệm:** Dịch vụ lưu trữ khối (block storage) hiệu năng cao, bền bỉ, được thiết kế để sử dụng với Amazon EC2 instances. Hoạt động giống như ổ đĩa mạng.
    - **Đặc điểm:** Gắn vào một EC2 instance cụ thể trong một Availability Zone (AZ). Dữ liệu tồn tại độc lập với vòng đời của instance. Cung cấp nhiều loại volume khác nhau (gp2/gp3, io1/io2, st1, sc1) tối ưu cho các workload khác nhau (SSD cho hiệu năng cao, HDD cho thông lượng). Hỗ trợ tạo **Snapshots** (bản sao lưu tại một thời điểm) và lưu trữ chúng trên S3 để tăng độ bền.
    - **Sử dụng:** Boot volumes cho EC2, lưu trữ cơ sở dữ liệu chạy trên EC2, lưu trữ cho các ứng dụng đòi hỏi truy cập block-level.
  + **Amazon S3 (Simple Storage Service):**
    - **Khái niệm:** Dịch vụ lưu trữ đối tượng (object storage) có khả năng mở rộng, độ bền (99.999999999% - 11 số 9) và tính sẵn sàng cao.
    - **Đặc điểm:** Lưu trữ dữ liệu dưới dạng objects trong các **Buckets** (tên bucket là duy nhất toàn cầu). Không giới hạn dung lượng lưu trữ. Truy cập qua API hoặc giao diện web. Cung cấp nhiều **Storage Classes** để tối ưu chi phí dựa trên tần suất truy cập và yêu cầu về độ bền/sẵn sàng (S3 Standard, S3 Intelligent-Tiering, S3 Standard-IA, S3 One Zone-IA, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval, S3 Glacier Deep Archive). Hỗ trợ các tính năng như Versioning, Lifecycle Policies, Static Website Hosting, Encryption.
    - **Sử dụng:** Lưu trữ và phân phối file (hình ảnh, video, tài liệu), backup và phục hồi dữ liệu, data lakes, static website hosting, lưu trữ dữ liệu ứng dụng.
  + **Amazon EFS (Elastic File System):**
    - **Khái niệm:** Dịch vụ lưu trữ file (file storage) đơn giản, có khả năng mở rộng, đàn hồi, được quản lý hoàn toàn cho các ứng dụng dựa trên Linux chạy trên AWS hoặc on-premises.
    - **Đặc điểm:** Cung cấp giao diện hệ thống file chuẩn (NFSv4). Có thể được mount (gắn) đồng thời vào hàng nghìn EC2 instances (hoặc các dịch vụ compute khác như ECS, EKS, Lambda) nằm trong nhiều Availability Zones khác nhau trong cùng một Region. Tự động tăng hoặc giảm dung lượng khi bạn thêm hoặc xóa file. Có các tùy chọn về hiệu năng (General Purpose, Max I/O) và thông lượng (Bursting, Provisioned).
    - **Sử dụng:** Lưu trữ nội dung web dùng chung cho nhiều web server, home directories cho người dùng, chia sẻ dữ liệu giữa các ứng dụng, big data analytics.
  + **Amazon S3 Glacier:**
    - **Khái niệm:** Dịch vụ lưu trữ đối tượng chi phí cực thấp, an toàn và bền bỉ cho việc lưu trữ dữ liệu dài hạn (data archiving) và sao lưu dài hạn. Đây là một nhóm các storage class của S3.
    - **Đặc điểm:** Được thiết kế cho dữ liệu ít khi truy cập. Chi phí lưu trữ rất rẻ nhưng thời gian truy xuất dữ liệu lâu hơn S3 Standard (từ vài phút đến vài giờ, tùy thuộc vào tùy chọn truy xuất: Expedited, Standard, Bulk). **S3 Glacier Deep Archive** là lớp lưu trữ rẻ nhất, dành cho dữ liệu truy cập 1-2 lần/năm, thời gian truy xuất thường là trong vòng 12 giờ. Thường được sử dụng kết hợp với S3 Lifecycle Policies để tự động chuyển dữ liệu cũ sang Glacier.
    - **Sử dụng:** Lưu trữ archive theo yêu cầu pháp lý hoặc tuân thủ, thay thế băng từ (tape libraries), lưu trữ dữ liệu media cũ, backup dài hạn.

**2. Tóm tắt nội dung video :**

Dựa trên tiêu đề các video trong hình ảnh:

* **Introduction Video:** Giới thiệu tầm quan trọng của việc lựa chọn đúng giải pháp lưu trữ, phân biệt các loại lưu trữ cơ bản (Block, Object, File) và giới thiệu các dịch vụ sẽ được học (EBS, S3, EFS, Glacier).
* **Section 1 Video - AWS EBS:** Đi sâu vào EBS: giải thích khái niệm block storage, các loại volume EBS (SSD vs HDD, General Purpose vs Provisioned IOPS), cách gắn volume vào EC2, và khái niệm snapshots để sao lưu.
* **Console Demonstrations - EBS:** Hướng dẫn thực hành trên AWS Console: tạo EBS volume, gắn vào EC2 instance, tạo file system trên volume, tạo snapshot từ volume, và có thể là khôi phục volume từ snapshot.
* **Section 2 Video - AWS S3:** Trình bày về S3: khái niệm object storage, buckets và objects, độ bền và tính sẵn sàng cao, các trường hợp sử dụng phổ biến, giới thiệu các lớp lưu trữ (Storage Classes) khác nhau và lợi ích của chúng. Có thể đề cập versioning và lifecycle policies.
* **Console Demonstration S3:** Hướng dẫn thực hành trên AWS Console: tạo S3 bucket, tải lên (upload) và tải xuống (download) object, cấu hình quyền truy cập cơ bản, có thể demo bật versioning hoặc tạo lifecycle rule đơn giản để chuyển dữ liệu sang lớp lưu trữ khác.
* **Section 3 Video - AWS EFS:** Giải thích về EFS: khái niệm file storage, giao thức NFS, khả năng chia sẻ giữa nhiều EC2 instance và AZs, tính đàn hồi (tự động co giãn), các chế độ hiệu năng và thông lượng.
* **Console Demonstrations - S3 and EFS:** Hướng dẫn thực hành trên AWS Console: tạo EFS file system, cấu hình mount target, mount EFS vào một hoặc nhiều EC2 instance (Linux), đọc/ghi file để chứng minh tính chia sẻ. Có thể so sánh cách truy cập EFS so với S3.
* **Section 4 Video - AWS S3 Glacier:** Tập trung vào S3 Glacier và Glacier Deep Archive: mục đích (archiving), chi phí cực thấp, các tùy chọn và thời gian truy xuất dữ liệu, cách sử dụng kết hợp với S3 Lifecycle Policies.
* **Console Demonstrations - Glacier:** Hướng dẫn thực hành trên AWS Console: cấu hình S3 Lifecycle rule để tự động chuyển object sang Glacier/Glacier Deep Archive, demo cách khởi tạo một yêu cầu khôi phục (restore request) object từ Glacier.
* **Wrap Up Video:** Tổng kết lại các dịch vụ lưu trữ đã học (EBS, S3, EFS, Glacier), nhấn mạnh sự khác biệt chính và các yếu tố cần cân nhắc khi lựa chọn giải pháp lưu trữ phù hợp cho từng nhu cầu cụ thể.

**3. Các Case Study :**

* **Case Study 1: Lưu trữ Hệ điều hành và Cơ sở dữ liệu (Sử dụng EBS)**
  + *Tình huống:* Một ứng dụng doanh nghiệp chạy trên EC2 instance, cần nơi lưu trữ hệ điều hành (boot volume) và một cơ sở dữ liệu quan hệ (ví dụ: MySQL, PostgreSQL) đòi hỏi hiệu năng I/O cao và độ trễ thấp.
  + *Giải pháp:* Sử dụng EBS gp3 (General Purpose SSD) hoặc io2 (Provisioned IOPS SSD) volume để làm boot volume và lưu trữ dữ liệu cho cơ sở dữ liệu. Thường xuyên tạo EBS snapshots để sao lưu.
  + *Lợi ích:* Hiệu năng cao, độ trễ thấp đáp ứng yêu cầu CSDL, dữ liệu bền bỉ và có thể sao lưu dễ dàng qua snapshots.
* **Case Study 2: Lưu trữ và Phân phối Nội dung Tĩnh (Sử dụng S3)**
  + *Tình huống:* Một trang web tin tức hoặc blog có lượng lớn hình ảnh, video và các file CSS/JS cần được lưu trữ và phân phối hiệu quả đến người dùng trên toàn cầu.
  + *Giải pháp:* Lưu trữ tất cả các file tĩnh này trong một S3 bucket. Cấu hình bucket cho phép truy cập công khai (public read) hoặc sử dụng S3 kết hợp với Amazon CloudFront (CDN) để tăng tốc độ phân phối và giảm độ trễ.
  + *Lợi ích:* Khả năng mở rộng lưu trữ gần như vô hạn, độ bền cao, chi phí hiệu quả, dễ dàng tích hợp với CDN.
* **Case Study 3: Chia sẻ Dữ liệu giữa các Máy chủ Web (Sử dụng EFS)**
  + *Tình huống:* Một hệ thống quản lý nội dung (CMS) hoặc một ứng dụng web được triển khai trên nhiều EC2 instance (để cân bằng tải và tăng tính sẵn sàng). Tất cả các instance này cần truy cập vào cùng một bộ dữ liệu chung (ví dụ: file tải lên của người dùng, mã nguồn dùng chung).
  + *Giải pháp:* Tạo một EFS file system và mount nó vào tất cả các EC2 instance trong cụm web server. Các instance có thể đọc/ghi dữ liệu vào EFS như một thư mục cục bộ chia sẻ.
  + *Lợi ích:* Dữ liệu nhất quán trên tất cả các instance, dễ dàng chia sẻ file, tự động mở rộng dung lượng lưu trữ.
* **Case Study 4: Lưu trữ Archive Dài hạn (Sử dụng S3 Glacier / Deep Archive)**
  + *Tình huống:* Một tổ chức tài chính cần lưu trữ hồ sơ giao dịch khách hàng trong 7 năm theo quy định. Dữ liệu này hiếm khi cần truy cập, nhưng phải được lưu trữ an toàn và có thể truy xuất khi cần kiểm toán.
  + *Giải pháp:* Sử dụng S3 Lifecycle Policies để tự động chuyển các bản ghi giao dịch từ S3 Standard sang S3 Glacier hoặc S3 Glacier Deep Archive sau một khoảng thời gian nhất định (ví dụ: 90 ngày).
  + *Lợi ích:* Giảm đáng kể chi phí lưu trữ cho dữ liệu cũ, đảm bảo tuân thủ quy định lưu trữ dài hạn, dữ liệu vẫn được bảo vệ an toàn và bền bỉ.

Module 8 : **Databases**

**1. Khái niệm :**

* **Tổng quan về Cơ sở dữ liệu trên AWS:**
  + Tầm quan trọng của CSDL trong ứng dụng hiện đại.
  + Những thách thức khi tự quản lý CSDL (cài đặt, vá lỗi, sao lưu, mở rộng, HA).
  + Lợi ích của việc sử dụng CSDL được quản lý trên AWS (giảm gánh nặng vận hành, dễ mở rộng, độ tin cậy cao, bảo mật tích hợp).
* **Phân loại Cơ sở dữ liệu:**
  + **CSDL Quan hệ (Relational - SQL):** Dữ liệu có cấu trúc dạng bảng, mối quan hệ rõ ràng, sử dụng SQL. Phù hợp cho OLTP (Online Transaction Processing), ứng dụng cần tính toàn vẹn dữ liệu cao (ACID).
  + **CSDL Phi quan hệ (Non-Relational - NoSQL):** Mô hình dữ liệu linh hoạt (Key-Value, Document, etc.), thường tối ưu cho khả năng mở rộng và hiệu năng cao ở quy mô lớn. Phù hợp cho dữ liệu phi/bán cấu trúc, ứng dụng web scale, IoT, game.
  + **Kho dữ liệu (Data Warehouse):** Tối ưu cho OLAP (Online Analytical Processing) - phân tích dữ liệu lớn, chạy các truy vấn phức tạp.
* **Các Dịch vụ Cơ sở dữ liệu chính trên AWS (chi tiết hơn):**
  + **Amazon RDS (Relational Database Service):**
    - **Khái niệm:** Dịch vụ quản lý CSDL quan hệ. AWS lo các tác vụ nền tảng.
    - **Database Engines hỗ trợ:** MySQL, PostgreSQL, MariaDB, Oracle, SQL Server.
    - **Tính năng chính:** Dễ dàng cài đặt & cấu hình, tự động vá lỗi (Managed Patching), tự động sao lưu (Automated Backups), **Multi-AZ Deployments** (Triển khai Đa Vùng sẵn sàng - cho High Availability, tự động failover), **Read Replicas** (Bản sao chỉ đọc - để mở rộng quy mô đọc).
    - **Sử dụng:** Ứng dụng web và mobile truyền thống, ứng dụng doanh nghiệp (ERP, CRM), ứng dụng cần CSDL quan hệ.
  + **Amazon DynamoDB:**
    - **Khái niệm:** Dịch vụ CSDL NoSQL (key-value và document) được quản lý hoàn toàn, **serverless**.
    - **Đặc điểm:** Khả năng mở rộng tự động (gần như vô hạn), hiệu năng cao với độ trễ mili giây đơn vị nhất quán ở mọi quy mô, độ bền và tính sẵn sàng cao được tích hợp sẵn, mô hình định giá linh hoạt (Provisioned Capacity hoặc On-Demand).
    - **Cấu trúc cơ bản:** Tables, Items (tương đương hàng), Attributes (tương đương cột). **Primary Key** (Khóa chính - Partition key, tùy chọn Sort key) để định danh duy nhất item.
    - **Sử dụng:** Ứng dụng web quy mô lớn, backend cho mobile, game (bảng xếp hạng, trạng thái người chơi), IoT, ứng dụng cần độ trễ thấp và khả năng mở rộng lớn, session store.
  + **Amazon Redshift:**
    - **Khái niệm:** Dịch vụ kho dữ liệu (Data Warehouse) đám mây, nhanh chóng, được quản lý hoàn toàn, quy mô petabyte.
    - **Đặc điểm:** Tối ưu cho phân tích dữ liệu (OLAP). Sử dụng **lưu trữ cột (columnar storage)** và **xử lý song song lớn (Massively Parallel Processing - MPP)** để thực thi các truy vấn phức tạp nhanh chóng trên tập dữ liệu lớn. Tích hợp tốt với các công cụ BI và S3 (cho Data Lake).
    - **Sử dụng:** Business Intelligence (BI), phân tích dữ liệu vận hành, dashboard báo cáo, phân tích log.
  + **Amazon Aurora:**
    - **Khái niệm:** CSDL quan hệ do AWS phát triển, **tương thích với MySQL và PostgreSQL**, được xây dựng cho đám mây. Là một tùy chọn engine trong RDS.
    - **Đặc điểm:** Cung cấp hiệu năng và tính sẵn sàng cao hơn đáng kể so với MySQL/PostgreSQL tiêu chuẩn (lên đến 5 lần thông lượng MySQL, 3 lần PostgreSQL). Hệ thống lưu trữ tự phục hồi, chịu lỗi, tự động mở rộng lên đến 128TB mỗi DB instance. Chi phí hiệu quả hơn các CSDL thương mại. Có các tính năng như Aurora Serverless, Global Database.
    - **Sử dụng:** Các ứng dụng đòi hỏi thông lượng và tính sẵn sàng cao hơn RDS thông thường, hiện đại hóa CSDL thương mại (Oracle/SQL Server), các ứng dụng quan hệ quy mô lớn.

**2. Tóm tắt nội dung video :**

* **Introduction Video:** Giới thiệu chung về module Databases, tầm quan trọng và các loại CSDL khác nhau sẽ được khám phá. Phân biệt SQL và NoSQL.
* **Section 1 Video - Amazon RDS:** Giải thích chi tiết về RDS, các engine hỗ trợ, lợi ích của việc sử dụng dịch vụ được quản lý. Tập trung vào các tính năng chính như Multi-AZ và Read Replicas.
* **Console Demonstration - RDS:** Hướng dẫn trực quan cách khởi chạy (launch) một RDS DB instance trên AWS Console, chọn engine, cấu hình kích thước, thiết lập thông tin đăng nhập, xem xét các tùy chọn Multi-AZ.
* **Section 2 Video - Amazon DynamoDB:** Trình bày về DynamoDB, giải thích nó là gì, tại sao lại là NoSQL, các khái niệm cốt lõi (Tables, Items, Keys), nhấn mạnh tính serverless, khả năng mở rộng và hiệu năng.
* **Console Demonstration - DynamoDB:** Hướng dẫn cách tạo một bảng DynamoDB trên Console, định nghĩa Primary Key, thêm (create) và đọc (read) items trong bảng.
* **Section 3 Video - Amazon Redshift:** Giới thiệu về Redshift như một giải pháp Data Warehouse. Giải thích các trường hợp sử dụng (OLAP, BI), cách nó hoạt động (columnar storage, MPP) và lợi ích so với việc dùng CSDL quan hệ thông thường cho phân tích.
* **Section 4 Video - Amazon Aurora:** Tập trung vào Aurora, giải thích sự tương thích với MySQL/PostgreSQL nhưng có hiệu năng và tính sẵn sàng vượt trội. Nêu bật kiến trúc lưu trữ độc đáo và các tính năng nâng cao của Aurora.
* **Wrap Up Video:** Tóm tắt các dịch vụ CSDL chính đã học (RDS, DynamoDB, Redshift, Aurora), nhấn mạnh sự khác biệt và khi nào nên sử dụng từng loại.

**3. Các Case Study :**

* **Case Study 1: Ứng dụng Web Thương mại Điện tử (Sử dụng RDS/Aurora)**
  + *Tình huống:* Một trang web bán hàng online cần lưu trữ thông tin sản phẩm, đơn hàng, khách hàng. Dữ liệu có cấu trúc rõ ràng, cần đảm bảo tính toàn vẹn giao dịch. Lưu lượng truy cập cao, đặc biệt vào mùa khuyến mãi.
  + *Giải pháp:* Sử dụng Amazon RDS với engine MySQL hoặc PostgreSQL. Kích hoạt Multi-AZ để đảm bảo tính sẵn sàng. Tạo Read Replicas để xử lý lượng truy cập đọc lớn. *Nâng cấp:* Nếu yêu cầu hiệu năng và độ tin cậy cực cao, có thể cân nhắc chuyển sang Amazon Aurora (tương thích MySQL/PostgreSQL).
  + *Lợi ích:* Quản lý CSDL dễ dàng, đảm bảo tính toàn vẹn dữ liệu, khả năng mở rộng tốt, tính sẵn sàng cao. Aurora cung cấp hiệu năng vượt trội cho các workload đòi hỏi cao.
* **Case Study 2: Ứng dụng Game Mobile với Bảng xếp hạng Toàn cầu (Sử dụng DynamoDB)**
  + *Tình huống:* Game mobile cần lưu profile người chơi, điểm số, vật phẩm, và một bảng xếp hạng cập nhật real-time với hàng triệu người chơi. Yêu cầu độ trễ thấp và khả năng xử lý lượng ghi/đọc cực lớn, biến động.
  + *Giải pháp:* Sử dụng Amazon DynamoDB. Thiết kế bảng với Primary Key phù hợp (ví dụ: GameID làm Partition Key, Score làm Sort Key cho bảng xếp hạng). Tận dụng khả năng mở rộng tự động (On-Demand capacity) hoặc cấu hình Provisioned Capacity.
  + *Lợi ích:* Độ trễ thấp ổn định, khả năng mở rộng gần như vô hạn, không cần quản lý server, phù hợp với dữ liệu bán cấu trúc và lượng truy cập biến đổi lớn.
* **Case Study 3: Hệ thống Phân tích Kinh doanh (Sử dụng Redshift)**
  + *Tình huống:* Doanh nghiệp thu thập dữ liệu bán hàng, marketing, log website từ nhiều nguồn. Bộ phận phân tích cần chạy các báo cáo phức tạp để tìm insight, phân tích xu hướng, tối ưu chiến dịch.
  + *Giải pháp:* Định kỳ ETL (Extract, Transform, Load) dữ liệu từ các nguồn (có thể là RDS, S3) vào Amazon Redshift. Sử dụng các công cụ BI (như Amazon QuickSight, Tableau) kết nối với Redshift để tạo dashboard và báo cáo phân tích.
  + *Lợi ích:* Xử lý truy vấn phân tích phức tạp nhanh chóng trên dữ liệu lớn, tối ưu cho BI và reporting, không ảnh hưởng hiệu năng hệ thống OLTP.
* **Case Study 4: Hiện đại hóa CSDL Doanh nghiệp (Sử dụng Aurora)**
  + *Tình huống:* Một công ty lớn đang chạy ứng dụng quan trọng trên CSDL Oracle hoặc SQL Server tự quản lý hoặc trên RDS. Họ muốn giảm chi phí bản quyền, tăng hiệu năng và giảm gánh nặng quản trị mà vẫn giữ được tính năng của CSDL quan hệ.
  + *Giải pháp:* Sử dụng AWS Database Migration Service (DMS) và Schema Conversion Tool (SCT) để di chuyển CSDL sang Amazon Aurora (phiên bản tương thích PostgreSQL hoặc MySQL).
  + *Lợi ích:* Giảm chi phí đáng kể so với CSDL thương mại, hiệu năng cao hơn, tính sẵn sàng tốt hơn, tận dụng các tính năng tối ưu cho cloud của Aurora.

Hy vọng phần tóm tắt chi tiết cho Module 8 này chính xác và hữu ích cho việc học của bạn!