# RESTful API ?

Lời nói đầu

Web Service là một ứng dụng tuyệt vời của kĩ thuật lập trình phân tán và được coi là một công nghệ mang đến cuộc cách mạng trong cách thức hoạt động của các dịch vụ B2B (Business to Business), B2C (Business to Customer) và nhiều loại dịch vụ mạng khác.

Khi các ứng dụng không liên quan tới nhau, nhưng vẫn muốn trao đổi dữ liệu với nhau thì người ta sẽ nghĩ ngay tới việc sử dụng web service. Một web service sẽ trả về dữ liệu theo một cấu trúc nào đó (XML hoặc JSON,…) để các ứng dụng khác có thể đọc, phân tích và sử dụng được.

Web service ra đời như là một điều hiển nhiên, bởi vì ngày càng có nhiều hệ thống chạy đa nền tảng như Facebook, Youtube,.. ra đời. Đặc điểm của các hệ thống chạy đa nền tảng này là luôn yêu cầu khả năng đồng bộ dữ liệu. Ví dụ bạn like một status facebook trên web, thì trên app cũng phải được thể hiện, bạn đăng một bức ảnh lên facebook từ mobile, thì trên web cũng phải nhìn thấy. Để làm được điều này, người ta sẽ tạo ra một con web service, để khi bạn đăng ảnh, like hay thực hiện bất kỳ hành động gì đều phải gọi tới web service này cho dù hành động đó được thực hiện từ web hay mobile. Mặt khác, ứng dụng web và mobile sẽ kết nối vào chung web service để đọc dữ liệu, vì vậy sẽ đảm bảo được dữ liệu là giống nhau cho dù trên các nền tảng khác nhau.

Mục đích của WebService:

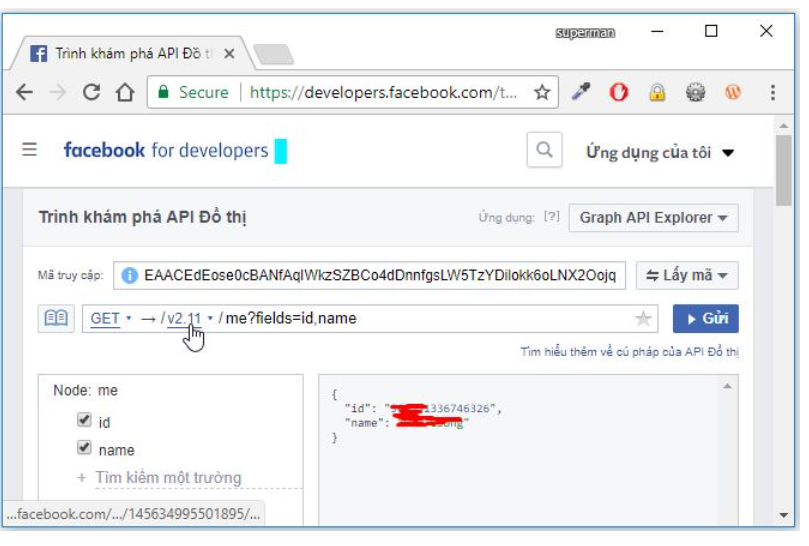
1. Giúp các hệ thống không liên quan tới nhau vẫn có thể giao tiếp được với nhau
2. Đồng bộ dữ liệu giữa các nền tảng

RESTful API là gì?

RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.

* Sử dụng các phương thức HTTP/HTTPs
* Phi trạng thái (stateless) (không lưu trữ thông tin của client)
* Hiển thị cấu trúc thư mục như các URls
* Truyền tải dữ liệu với định dạng phong phú: html, json, text, xml… (thường sử dụng nhiều với json)

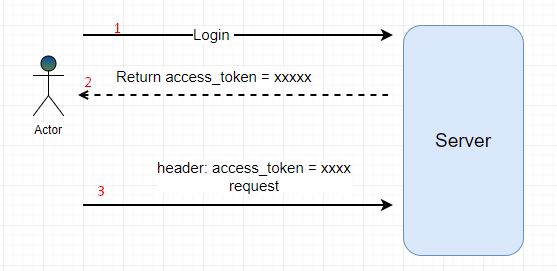
Ví dụ API của Facebook cũng là 1 dạng RESTful Web Service



### Bảo mật, xác thực trong restful web service

Request trong web service là stateless, nó cũng không sử dụng cookies nên ta không thể lưu thông tin đăng nhập vào session.

Để xác thực trong RESTful web service, sau khi đăng nhập client sẽ nhập được 1 token, mỗi lần gửi request sẽ gửi kèm token đó vào header của request. Server sẽ dựa vào token để xác thực client.



Nếu là web developer, chắc chắn bạn biết đến method GET và POST. Nhưng với RESTful thì có thêm một số method mới, kèm cách sử dụng tương ứng như sau:

* GET: được sử dụng để lấy thông tin từ sever theo URI đã cung cấp.
* POST: gửi thông tin tới sever thông qua các biểu mẫu http (đăng kí chả hạn..)
* HEAD: giống với GET nhưng response trả về không có body, chỉ có header
* PUT: ghi đè tất cả thông tin của đối tượng với những gì được gửi lên
* PATCH: ghi đè các thông tin được thay đổi của đối tượng.
* DELETE: xóa tài nguyên trên server.
* CONNECT: thiết lập một kết nối tới server theo URI.
* OPTIONS: mô tả các tùy chọn giao tiếp cho resource.
* TRACE: thực hiện một bài test loop – back theo đường dẫn đến resource.

## Status code

Khi chúng ta request một API nào đó thường thì sẽ có vài status code để nhận biết sau:

* 200 OK – Trả về thành công cho những phương thức GET, PUT, PATCH hoặc DELETE.
* 201 Created – Trả về khi một Resouce vừa được tạo thành công.
* 204 No Content – Trả về khi Resource xoá thành công.
* 304 Not Modified – Client có thể sử dụng dữ liệu cache.
* 400 Bad Request – Request không hợp lệ
* 401 Unauthorized – Request cần có auth.
* 403 Forbidden – bị từ chối không cho phép.
* 404 Not Found – Không tìm thấy resource từ URI
* 405 Method Not Allowed – Phương thức không cho phép với user hiện tại.
* 410 Gone – Resource không còn tồn tại, Version cũ đã không còn hỗ trợ.
* 415 Unsupported Media Type – Không hỗ trợ kiểu Resource này.
* 422 Unprocessable Entity – Dữ liệu không được xác thực
* 429 Too Many Requests – Request bị từ chối do bị giới hạn

Ưu điểm của RESTFUL API:

* Dễ viết và điều chỉnh hơn so với framework WCF, Web service
* Nâng cao trải nghiệm người dùng với UX thân thiện
* Thỏa mãn các tiêu chuẩn phổ biến về REST và HTTP, dễ sử dụng, dễ hiểu và thân thiện với các nhà phát triển.
* Mã nguồn mở, hỗ trợ đầy đủ chức năng RESTful, dùng cho tất cả client hỗ trợ XML, JSON.
* Hỗ trợ tốt các tính năng và thành phần của HTTP như: caching, versioning, HttpRequestMessage, HttpResponseMessage,…, web API MVC như: routing, container, controller, action result, filter, model binder, dependency injection, unit test.
* Tính bảo mật cao, xác nhận 2 chiều khi có request
* Dữ liệu trả về với nhiều định dạng khác nhau như: XML, JSON
* Ứng dụng đa dạng trên desktop, mobile và website.

REST đã nổi lên trong vài năm qua như là một mô hình thiết kế dịch vụ chiếm ưu thế. Trong thực tế, REST đã có những ảnh hưởng lớn và gần như thay thế SOAP và WSDL vì nó đơn giản và dễ sử dụng hơn rất nhiều. Với REST hệ thống không chỉ dễ mở rộng, dễ bảo trì mà còn linh động trong việc cung cấp các kiểu ứng dụng với các dữ liệu khác nhau.

**Ngôn ngữ JavaScript**

**Tổng quan về ngôn ngữ JavaScript [9]**

Ngôn ngữ lập trình Javascript được giới thiệu đầu tiên vào năm 1995. Mục đích là để đưa những chương trình vào trang web ở trình duyệt Netscape Navigator – một trình duyệt web phổ biến những năm 1990.JavaScript được phát triển bởi Brendan Eich tại Hãng truyền thông Netscape với cái tên đầu tiên là Mocha, rồi sau đó đổi tên thành LiveScript, và cuối cùng thành JavaScript. Có lẽ việc đổi tên như vậy là để giúp JavaScript được chú ý nhiều hơn. Bởi tại thời điểm này, Java đang được coi là một hiện tượng và trở nên phổ biến.

**JavaScript là gì?**

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (smalland lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

* *Client-side JavaScript* -JavaScript phía máy khách, JavaScript được mở rộngbằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.
* *Server-side JavaScript* -JavaScript phía Server, JavaScript được mở rộng bằngcách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.



Logo ngôn ngữ JavaScript

**API jQuery AJAX [11]**

**Tổng quan về AJAX**

Ajax = Asynchoronous JavaScript và XML

Ajax là một bộ công cụ cho phép load dữ liệu từ server mà không yêu cầu tải lại trang. Nó sử dụng chức năng sẵn có XMLHttpRequest (XHR) của trình duyệt để thực hiện một yêu cầu đến server và xử lý dữ liệu server trả về.

Một số ứng dụng sử dụng Ajax như: Gmail, Google Maps, Youtube, Facebook…

**Ngôn Ngữ Java:**

Java là một ngôn ngữ lập trình hướng đối tượng, dựa trên lớp được thiết kế để có càng ít phụ thuộc thực thi càng tốt. Nó là ngôn ngữ lập trình có mục đích chung cho phép các nhà phát triển ứng dụng viết một lần, chạy ở mọi nơi ,[9] nghĩa là mã Java đã biên dịch có thể chạy trên tất cả các nền tảng hỗ trợ Java mà không cần biên dịch lại. Các ứng dụng Java thường được biên dịch thành bytecode có thể chạy trên bất kỳ máy ảo Java (JVM) nào bất kể kiến trúc máy tính bên dưới. Cú pháp của Java tương tự như C và C++, nhưng có ít cơ sở cấp thấp hơn các ngôn ngữ trên.

Ngôn ngữ Java có cú pháp rất giống với ngôn ngữ C/C (là một ngôn ngữ rất mạnh được dùng phổ biến hiện nay), tuy nhiên nó đã được thay đổi khá nhiều để đáp ứng khả năng độc lập với hệ điều hành. Nền tảng của ngôn ngữ Java là các class. Các class đóng vai trò như những đối tượng, người lập trình khi xây dựng ứng dụng sẽ sử dụng một số class chuẩn của hệ thống, đồng thời có thể tự mình xây dựng class khác đáp ứng yêu cầu công việc. Java là ngôn ngữ dễ học, nó lược bớt những lệnh thừa và rườm rà của C/C để người lập trình chú trọng vào việc viết chương trình. Đồng thời, nó còn hạn chế người lập trình không can thiệp quá sâu vào hệ thống. Hiện nay Java được ứng dụng rất rộng rãi, chủ yếu là các ứng dụng được viết trên Internet nhằm khai thác tối đa khả năng của nó trong lĩnh vực viễn thông, truyền hình, mạng, và các máy tính đơn lẻ. Hiện tại các ứng dụng viết bằng Java khi thực hiện còn chậm hơn so với các ứng dụng được xây dựng thông thường, tuy nhiên hầu hết các nhà phát triển phần mềm đều cho rằng tốc độ sẽ được cải thiện và Java sẽ là ngôn ngữ lập trình của tương lai.

## ****Đặc điểm của ngôn ngữ lập trình Java****

Là một ngôn ngữ lập trình hướng đối tượng vì vậy Java cũng có 4 đặc điểm chung sau đây:

****Tính trừu tượng (Abstraction)****: là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.

****Tính đa hình (Polymorphism)****: cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.

****Tính kế thừa (Inheritance)****: Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.

****Tính đóng gói (Encapsulation)****:  là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

Ngoài ra Java còn có một số đặc điểm sau:

****Độc lập nền (Write Once, Run Anywhere)****: Không giống như nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã bytecode chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.

****Đơn giản****: học Java thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++.

****Bảo mật****: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)…

****Thông dịch****: Java là một ngôn ngữ lập trình vừa biên dịch vừa thông dịch. Chương trình nguồn viết bằng ngôn ngữ lập trình Java có đuôi \*.java và được biên dịch thành tập tin có đuôi \*.class sau đó được trình thông dịch thông dịch thành mã máy.

****Đa luồng****: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được sử dụng rất nhiều trong lập trình game.

****Hướng đối tượng****: Hướng đối tượng trong Java tương tự như C++ nhưng Java là một ngôn ngữ lập trình hướng đối tượng hoàn toàn. Tất cả mọi thứ đề cập đến trong Java đều liên quan đến các đối tượng được định nghĩa trước, thậm chí hàm chính của một chương trình viết bằng Java (đó là hàm main) cũng phải đặt bên trong một lớp. Hướng đối tượng trong Java không có tính đa kế thừa (multi inheritance) như trong C++ mà thay vào đó Java đưa ra khái niệm interface để hỗ trợ tính đa kế thừa.

****Hiệu suất cao****:Nhờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.

****Linh hoạt****: Java được xem là linh hoạt hơn C/C ++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.

**Ưu Điểm Của Java:**

•    Dễ học hơn c ++  
•    Dễ đọc mã hơn c ++  
•    Phát triển nhanh hơn và có khả năng ít lỗi hơn (do quản lý bộ nhớ ít hơn) so với c ++.  
•    Hầu hết các nền tảng chéo kể từ khi nó được thiết kế cho nó và Sun tạo ra JRE và JDK cho hầu hết các nền tảng  
•    Tất cả mọi thứ được truyền bằng tham chiếu ngoại trừ các loại giá trị, theo mặc định  
•    Công cụ thu gom rác dọn dẹp đồ vật một khi chúng không còn được sử dụng, vì vậy bạn không phải tự mình theo dõi mọi thứ  
•    Rõ ràng java 6 chạy (một số?) Công cụ nhanh hơn một chút so với c ++ (mã kiểm tra được chuyển từ c ++ chạy nhanh hơn mặc dù phải làm những việc có vẻ lãng phí!)  
•    Chương trình được biên dịch và tối ưu hóa cho nền tảng, nếu cần?  
•    Dễ dàng tạo nhiều luồng và bạn có thể khai báo các chức năng được đồng bộ hóa để chỉ một luồng có thể có trong một luồng (IIRC)  
•    Bạn có thể sử dụng Wait () và notifyAll () và thông báo () để tạm dừng các luồng để chờ tín hiệu và đánh thức chúng lại.  
•    Các IDE miễn phí khác nhau có sẵn trên tất cả các nền tảng, chẳng hạn như Eclipse  
•    Nếu chương trình gặp sự cố, nó sẽ báo cho người dùng biết mã trong đó ở đâu và tại sao, hiển thị dấu vết ngăn xếp bao gồm số dòng (nếu biểu tượng gỡ lỗi được bật hoặc một cái gì đó tương tự) hoặc có thể được đổ vào tệp nhật ký lỗi.  
•    Các chương trình có thể được đóng gói thành các tệp jar với các tệp dữ liệu khác nhau trong các tệp jar khác nhau và được người dùng truy cập bằng Java WebStart, sẽ tự động cập nhật chương trình và mọi tệp jar cần thiết khác lên phiên bản mới nhất, chỉ tải xuống các tệp jar đã thay đổi.  
•    Mạng dễ dàng hơn / nhiều nền tảng hơn c ++.  
•    Các tệp được đọc và ghi dưới dạng big endian trên tất cả các nền tảng (là thứ tự byte mạng) đảm bảo rằng giao tiếp giữa các chương trình java và tải tệp không bao giờ yêu cầu hoán đổi thứ tự byte (trừ khi nói chuyện với chương trình không phải java hoặc sử dụng tệp được viết bởi một ).

## ****Ứng dụng của JAVA****

### **1. Java làm ứng dụng Android**

### **2. Java làm hệ thống giao dịch trong ngành Dịch vụ tài chính**

### **3. Java làm ứng dụng web**

### **4. Java làm phần mềm phát triển**

### 5. Java làm ứng dụng giao dịch

### **6. Java làm ứng dụng J2ME**

### **7. Java làm lập trình nhúng**

### **8. Java trong công nghệ Big Data**

### **9. Java làm hệ thống hiệu suất cao**

### **10. Java làm ứng dụng khoa học**

**Tìm hiểu API jQuery AJAX**

jQuery cung cấp một số phương thức để thực hiện các chức năng ajax. Chúng ta có thể yêu cầu các text, HTML, XML và JSON từ server sử dụng cả giao thức HTTP GET và HTTP POST, ngoài ra cũng có thể lấy dữ liệu từ bên ngoài trực tiếp vào trong phần tử được chọn.

* Phương thức jQuery **load()**:

Phương thức **load()** lấy dữ liệu từ server và trả dữ liệu cho phần tử được chọn.

|  |  |
| --- | --- |
| Cú pháp: | $(selector).load(URL,data,callback); |
|  |  |

* URL: required, đường dẫn đến file cần lấy thông tin.
* Data: cặp key/value gửi đi cùng với yêu cầu.
* Callback: tên của hàm sẽ được thực thi sau khi phương thức load hoàn thành.
* Phương thức **Post** trong jQuery Ajax

Có tác dụng lấy dữ liệu từ server bằng phương thức HTTP POST REQUEST

Cú pháp: $(selector).post(URL, data, function(data, status, xhr), dataType)

* URL: required, đường dẫn đến file cần lấy thông tin.
* data: không bắt buộc, là một đối tượng object gồm các key : value sẽ gửi lên server.
* function(data, status, xhr): là function sẽ xử lý khi thực hiện thành công với các parameters:

o data : bao gồm các dữ liệu trả về từ request

o status : gồm trạng thái request (“success” , “notmodified” , “error” , “timeout” or “parsererror”)

o xhr : gồm các đối tượng XMLHttpRequest

* dataType: là dạng dữ liệu trả về (text, json, script, xml, html, jsonp).
* Phương thức **Get** trong jQuery Ajax

Là phương pháp lấy dữ liệu từ server bằng phương thức HTTP GET Tương tự như phương thức Post, phương thức get có cú pháp là:

$.get(URL, data, function(data, status, xhr), dataType)

* URL: required, đường dẫn đến file cần lấy thông tin.
* data: không bắt buộc, là một đối tượng object gồm các key : value sẽ gửi lên server.
* function(data, status, xhr): là function sẽ xử lý khi thực hiện thành công với các parameters:

o data : bao gồm các dữ liệu trả về từ request

o status : gồm trạng thái request (“success” , “notmodified” , “error” , “timeout” or “parsererror”)

o xhr : gồm các đối tượng XMLHttpRequest

* dataType: là dạng dữ liệu trả về (text, json, script, xml, html, jsonp).

## MySQL là gì?

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc ( SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux , UNIX và Windows. MySQL thường được kết hợp với các ứng dụng web.



SQL là ngôn ngữ phổ biến nhất để thêm, truy cập và quản lý nội dung trong cơ sở dữ liệu. Nó được chú ý nhất vì khả năng xử lý nhanh, độ tin cậy đã được chứng minh, dễ sử dụng và linh hoạt. MySQL là một phần thiết yếu của hầu hết mọi ứng dụng [PHP](https://viettuts.vn/php) mã nguồn mở. Các ví dụ điển hình cho các tập lệnh dựa trên PHP và MySQL là WordPress, Joomla, Magento và Drupal. MySQL đang trở nên phổ biến vì nhiều lý do tốt:

* MySQL được phát hành theo giấy phép nguồn mở. Vì vậy, bạn không phải trả tiền để sử dụng nó.
* MySQL là một chương trình rất mạnh theo đúng nghĩa của nó. Nó xử lý một tập hợp lớn các chức năng của các gói cơ sở dữ liệu mạnh mẽ và đắt tiền nhất.
* MySQL sử dụng một dạng chuẩn của ngôn ngữ dữ liệu SQL nổi tiếng.
* MySQL hoạt động trên nhiều hệ điều hành và với nhiều ngôn ngữ bao gồm PHP, PERL, C, C ++, JAVA, v.v.
* MySQL hoạt động rất nhanh và hoạt động tốt ngay cả với các tập dữ liệu lớn.
* MySQL rất thân thiện với PHP, ngôn ngữ được đánh giá cao nhất để phát triển web.
* MySQL hỗ trợ cơ sở dữ liệu lớn, lên tới 50 triệu hàng hoặc nhiều hơn trong một bảng. Giới hạn kích thước tệp mặc định cho một bảng là 4GB, nhưng bạn có thể tăng mức này (nếu hệ điều hành của bạn có thể xử lý nó) đến giới hạn lý thuyết là 8 triệu terabyte (TB).
* MySQL là tùy biến. Giấy phép GPL mã nguồn mở cho phép các lập trình viên sửa đổi phần mềm MySQL để phù hợp với môi trường cụ thể của riêng họ.

## Cách thức hoạt động của MySQL

MySQL dựa trên mô hình client-server. Cốt lõi của MySQL là máy chủ MySQL, xử lý tất cả các hướng dẫn cơ sở dữ liệu (hoặc các lệnh). Máy chủ MySQL có sẵn như là một chương trình riêng biệt để sử dụng trong môi trường mạng client-server và như một thư viện có thể được nhúng (hoặc liên kết) vào các ứng dụng riêng biệt.

MySQL hoạt động cùng với một số chương trình tiện ích hỗ trợ quản trị cơ sở dữ liệu MySQL. Các lệnh được gửi đến MySQLServer thông qua máy khách MySQL, được cài đặt trên máy tính.

MySQL ban đầu được phát triển để xử lý cơ sở dữ liệu lớn một cách nhanh chóng. Mặc dù MySQL thường chỉ được cài đặt trên một máy, nhưng nó có thể gửi cơ sở dữ liệu đến nhiều vị trí, vì người dùng có thể truy cập thông qua các giao diện máy khách MySQL khác nhau. Các giao diện này gửi các câu lệnh SQL đến máy chủ và sau đó hiển thị kết quả.

## Các tính năng cốt lõi của MySQL

MySQL cho phép dữ liệu được lưu trữ và truy cập trên nhiều công cụ lưu trữ, bao gồm InnoDB, CSV và NDB. MySQL cũng có khả năng sao chép dữ liệu và phân vùng bảng để có hiệu suất và độ bền tốt hơn. Người dùng MySQL không bắt buộc phải học các lệnh mới; họ có thể truy cập dữ liệu của mình bằng các lệnh [SQL](https://viettuts.vn/sql) tiêu chuẩn.

MySQL được viết bằng C và C++ và có thể truy cập và có sẵn trên hơn 20 nền tảng, bao gồm Mac, Windows, Linux và Unix. RDBMS hỗ trợ cơ sở dữ liệu lớn với hàng triệu bản ghi và hỗ trợ nhiều loại dữ liệu bao gồm các số nguyên có chữ ký hoặc không dấu có độ dài 1, 2, 3, 4 và 8 byte(s); FLOAT; DOUBLE; CHAR; VARCHAR; BINARY; VARBINARY; TEXT; BLOB; DATE; TIME; DATETIME; TIMESTAMP; YEAR; SET; ENUM; và các kiểu OpenGIS. Các loại chuỗi có độ dài cố định và biến đổi cũng được hỗ trợ.

Để bảo mật, MySQL sử dụng một đặc quyền truy cập và hệ thống mật khẩu được mã hóa cho phép xác minh dựa trên máy chủ. Các máy khách MySQL có thể kết nối với Máy chủ MySQL bằng một số giao thức, bao gồm cả giao thức TCP/IP trên bất kỳ nền tảng nào. MySQL cũng hỗ trợ một số chương trình máy khách và tiện ích, chương trình dòng lệnh và công cụ quản trị như MySQL Workbench.

### ****Ưu điểm của MySQL****



* ***Dễ sử dụng***: ****MySQL**** là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
* ***Độ*** ***bảo mật cao***:  ****MySQL**** rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.
* ***Đa tính năng***: ****MySQL**** hỗ trợ rất nhiều ****chức năng SQL**** được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
* ***Khả năng mở rộng và mạnh mẽ***: ****MySQL**** có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
* ***Nhanh chóng***: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

**Spring Framwork:**

Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lơn.

Spring Framework được xây dựng dựa trên 2 nguyên tắc design chính là: Dependency Injection và Aspect Oriented Programming.

Những tính năng core (cốt lõi) của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object)

Spring là gì? Giới thiệu Spring Framework trong Java

# **2. Kiến trúc, các module của Spring Framework**

Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó.

Dưới đây là kiến trúc tổng thể của Spring Framework.

## **Test**

Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.

## **Spring Core Container**

Bao gồm các module spring core, beans, context và expression languate (EL)

* Spring core, bean cung cấp tính năng IOC và Dependency Injection.
* Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB, JMX.
* Expression Language được mở rộng từ Expresion Language trong JSP. Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

## **AOP, Aspects and Instrumentation**

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

## **Data Access / Integration**

Nhóm này bao gồm JDBC, ORM, OXM, JMS và module Transaction. Những module này cung cấp khả năng giao tiếp với database

## **Web**

Hay còn gọi là Spring MVC Nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

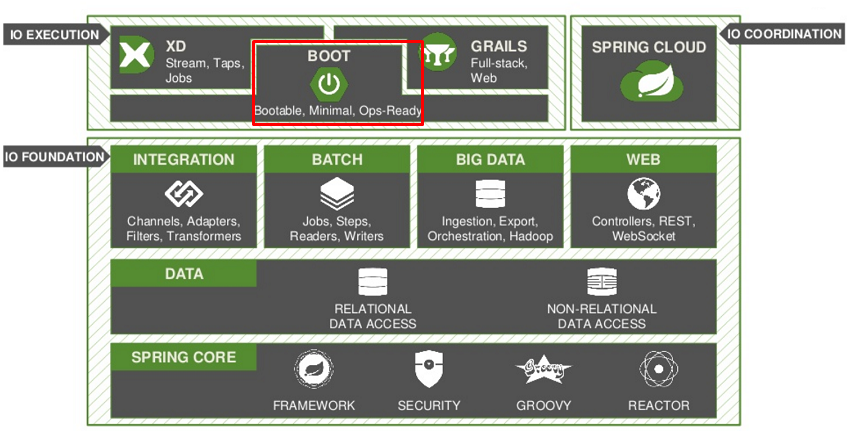
Spring là gì? Giới thiệu Spring Framework trong Java

# **3. Các lợi ích của Spring Framework.**

* Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.
* Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại.
* Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, JEE, các thư viện tạo lịch trình (Quartz và JDK timer)…
* Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.

**Spring Boot:**

* Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development) – Phát triển ứng dụng nhanh.
* Spring Boot được dùng để tạo các ứng dụng độc lập dựa trên Spring.
* Spring Boot không yêu cầu cấu hình XML
* Nó là một chuẩn cho cấu hình thiết kế phần mềm, tăng cao năng suất cho developer.

[](https://stackjava.com/wp-content/uploads/2018/02/spring-boot-architecture.jpg.png)

## **2. Ưu điểm của Spring Boot.**

* Có các tính năng của Spring Framework.
* Tạo ứng dụng độc lập, có thể chạy bằng java -jar (cho cả java web)
* Nhúng trực tiếp các ứng dụng server (Tomcat, Jetty…) do đó không cần phải triển khai file WAR
* Cấu hình ít, tự động cậu hình bất kì khi nào có thể (Giảm thời gian viết code, tăng năng suất)
* Không yêu cầu XML config…
* Cung cấp nhiều plugin
* Chuẩn cho Microservices (Cloud support; giảm việc setup, config; các thư viện hỗ trợ…)

**Spring Security:**

Khi làm việc với các ứng dụng cần bảo mật thì bắt buộc các bạn phải biết đến 2 khái niệm authentication và authorization. Nói nôm na thì authentication liên quan đến việc ai được quyền sử dụng ứng dụng còn authorization liên quan đến việc những resource nào, những phần nào của ứng dụng mà các bạn được phép sử dụng. Đối với các ứng dụng web có sử dụng Spring framework thì các bạn có thể sử dụng Spring Security để hiện thực 2 khái niệm này.

Spring Security là một khung công tác mạnh mẽ và có thể tùy chỉnh cao giúp xác thực và ủy quyền (hoặc kiểm soát truy cập); nó là mô-đun mặc định để bảo mật các ứng dụng Spring. Sau đây là một số tính năng quan trọng.

* Tích hợp API Servlet
* Tích hợp với Spring Web MVC và WebFlux
* Tích hợp với Spring Web MVC và WebFlux
* Bảo vệ chống lại các cuộc tấn công như cố định phiên, tấn công bằng nhấp chuột, CSRF  
  (giả mạo yêu cầu trên nhiều trang web), CORS (chia sẻ tài nguyên nguồn gốc chéo),  
  v.v.
* Hỗ trợ mở rộng và toàn diện cho cả xác thực và  
  ủy quyền
* Tích hợp với các công nghệ này: HTTP Basic, HTTP Digest, X.509,  
  LDAP, Form-based, OpenID, CAS, RMI, Kerberos, JAAS, Java EE, và  
  nhiều hơn nữa
* Tích hợp với các công nghệ của bên thứ ba: AppFuse, DWR, Grails,  
  Tapestry, JOSSO, AndroMDA, Roller, v.v.

Bảo mật của một ứng dụng web xoay quanh ba khái niệm chính:  
• Xác thực  
• Ủy quyền  
• Mã hóa

AWS S3

**[Amazon](https://cuongquach.com/tag/amazon" \t "https://cuongquach.com/_blank)**Simple Storage Service (Amazon S3)**** là một dịch vụ lưu trữ đối tượng cung cấp khả năng thay đổi theo quy mô, tính khả dụng của dữ liệu, bảo mật và hiệu năng hàng đầu trong lĩnh vực. Điều này có nghĩa là khách hàng thuộc mọi quy mô và lĩnh vực có thể sử dụng dịch vụ này để lưu trữ và bảo vệ bất kỳ lượng dữ liệu nào cho nhiều trường hợp sử dụng khác nhau, chẳng hạn như trang web, ứng dụng di động, sao lưu và khôi phục, lưu trữ, ứng dụng doanh nghiệp, thiết bị IoT và phân tích dữ liệu hớn.

****Amazon S3**** cung cấp các tính năng quản lý dễ sử dụng, nhờ đó, bạn có thể tổ chức dữ liệu và cấu hình các kiểm soát truy cập được tinh chỉnh để đáp ứng yêu cầu cụ thể của doanh nghiệp, tổ chức và yêu cầu về tuân thủ.****Amazon S3**** được thiết kế để đảm bảo độ bền 99,999999999% (11,9’s downtime) và lưu trữ dữ liệu của hàng triệu ứng dụng cho các công ty trên toàn thế giới.

****Amazon S3**** được thiết kế các tính năng tối giản nhằm đạt được mục tiêu đơn giản và nhanh gọn. Sau đây là một số ưu điểm của Amazon S3:

* ****Tạo Buckets****: tạo và đặt tên một Bucket sẽ chứa dữ liệu. Bucket là khái niệm cơ bản trong Amazon S3 dành cho việc lưu trữ dữ liệu.
* ****Lưu trữ dữ liệu trong Bucket**** : bạn có thể lưu trữ vô hạn các loại dữ liệu khác nhau trong một bucket. Mỗi dữ liệu của bạn sẽ như là một Object và muốn up nhiêu cũng được. Mỗi object có thể chứa tới 5TB dữ liệu.
* ****Tải dữ liệu :**** bạn có thể tải dữ liệu của bạn về bất cứ lúc nào, thậm chí là cho phép người khác tải dữ liệu của bạn đang nằm trong Amazon S3 Bucket nữa cơ.
* ****Phân quyền**** : phân quyền cho phép hoặc từ chối quyền hạn upload/download đối với dữ liệu nằm trong Amazon S3 Bucket của bạn.
* ****Giao diện tương tác tiêu chuẩn**** : bạn có thể sử dụng REST hoặc SOAP để thiết kế tương tác từ ứng dụng của bạn đến Amazon S3 qua các công cụ lập trình phát triển.
* ****Tính ổn định****: Nó được thiết kế để chịu được các hỏng hóc và phục hồi hệ thống rất nhanh với thời gian tối thiểu. Amazon cung cấp một thỏa thuận cấp dịch vụ (service-level agreement – SLA) để duy trì tính sẵn sàng ở mức 99.99 phần trăm.
* ****Đơn giản****: Dễ dùng S3 được xây dụng trên các khái niệm đơn giản và cung cấp tính mềm dẻo cao cho việc phát triển các ứng dụng của bạn. Bạn có thể xây dựng các lược đồ lưu trữ phức tạp hơn, nếu cần, bằng cách thêm các hàm vào các thành phần của S3.
* ****Tính mở rộng****: Thiết kế của S3 cung cấp một cấp độ cao về tính mở rộng và cho phép sự điều chỉnh dễ dàng trong dịch vụ khi lượng truy cập vào ứng dụng web của bạn tăng đột biến với lưu lượng khổng lồ.
* ****Rất Rẻ****: Chi phí sử dụng S3 rất cạnh tranh với các giải pháp của công ty và cá nhân khác trên thị trường.

## **Lợi ích của dịch vụ Amazon S3**

* ****Hiệu năng, khả năng thay đổi theo quy mô, tính khả dụng và độ bền hàng đầu trong lĩnh vực****

Tăng và giảm quy mô tài nguyên lưu trữ của bạn để đáp ứng nhu cầu hay thay đổi, mà không cần các khoản đầu tư trả trước hay chu kỳ thu mua tài nguyên. Amazon S3 được thiết kế để đảm bảo độ bền dữ liệu 99,999999999% (11 9’s) vì dịch vụ này tự động tạo và lưu trữ bản sao của tất cả đối tượng S3 trên nhiều hệ thống. Điều này có nghĩa là dữ liệu của bạn khả dụng khi cần và được bảo vệ khỏi các sự cố, lỗi và mối đe dọa.

* ****Nhiều lớp lưu trữ tiết kiệm chi phí****

Tiết kiệm chi phí mà không phải hi sinh hiệu suất bằng cách lưu trữ dữ liệu trên Các lớp lưu trữ S3 hỗ trợ các cấp độ truy cập dữ liệu khác nhau ở mức giá tương ứng. Bạn có thể sử dụng Phân tích lớp lưu trữ S3 để khám phá dữ liệu cần di chuyển sang lớp lưu trữ có chi phí thấp hơn dựa trên các cấu trúc truy cập và cấu hình một chính sách Vòng đời S3 để thực thi việc truyền dữ liệu. Bạn cũng có thể lưu trữ dữ liệu với các cấu trúc truy cập không xác định hoặc hay thay đổi trong S3 Thông minh-Phân bậc, dịch vụ này phân bậc đối tượng dựa trên các cấu trúc truy cập hay thay đổi và tự động tiết kiệm chi phí.

* ****Bảo mật, tính tuân thủ và khả năng kiểm tra chưa từng có****

Lưu trữ dữ liệu của bạn trong Amazon S3 và ngăn chặn truy cập dữ liệu trái phép với các tính năng mã hóa và công cụ quản lý truy cập. Bạn cũng có thể sử dụng Amazon Macie để xác định dữ liệu nhạy cảm trong bộ chứa S3 của mình và phát hiện yêu cầu truy cập bất thường. Amazon S3 duy trì các chương trình tuân thủ, chẳng hạn như PCI-DSS, HIPAA/HITECH, FedRAMP, Chỉ thị bảo vệ dữ liệu của Liên minh châu Âu và FISMA, để giúp bạn đáp ứng các yêu cầu về quy định. AWS cũng hỗ trợ nhiều khả năng kiểm tra để giám sát các yêu cầu truy cập vào tài nguyên S3 của bạn.

* ****Các công cụ quản lý dùng để kiểm soát dữ liệu chi tiết****

Phân loại, quản lý và báo cáo về dữ liệu của bạn bằng các tính năng như: Phân tích lớp lưu trữ S3 để phân tích các cấu trúc truy cập; các chính sách Vòng đời S3 để chuyển đối tượng sang lớp lưu trữ có chi phí thấp hơn; Sao chép liên khu vực S3 để sao chép dữ liệu vào khu vực khác; Khóa đối tượng S3 để áp dụng ngày duy trì đối với ứng dụng và bảo vệ để ứng dụng không bị xóa; và Kho S3 để tìm hiểu về đối tượng được lưu trữ của bạn, siêu dữ liệu và trạng thái mã hóa của các đối tượng đó. Bạn cũng có thể sử dụng S3 Batch Operations để thay đổi thuộc tính đối tượng và thực hiện tác vụ quản lý lưu trữ cho hàng tỷ đối tượng. Do Amazon S3 có liên kết với AWS Lambda, bạn có thể ghi nhật ký hoạt động, định nghĩa cảnh báo và tự động hóa luồng công việc mà không phải quản lý thêm cơ sở hạ tầng nào.

* ****Dịch vụ truy vấn tại chỗ cho phân tích****

Chạy phân tích dữ liệu lớn trên các đối tượng S3 của bạn (và các tập hợp dữ liệu khác trong AWS) với dịch vụ truy vấn tại chỗ của chúng tôi. Sử dụng Amazon Athena để truy vấn dữ liệu S3 với các biểu thức SQL tiêu chuẩn và Amazon Redshift Spectrum để phân tích dữ liệu được lưu trữ trên kho lưu trữ dữ liệu AWS và tài nguyên S3 của bạn. Bạn cũng có thể sử dụng S3 Select để truy xuất các tập hợp nhỏ siêu dữ liệu đối tượng thay vì toàn bộ đối tượng và tăng tới 400% hiệu suất truy vấn.

* ****Dịch vụ lưu trữ đám mây được hỗ trợ nhiều nhất****

Lưu trữ và bảo vệ dữ liệu của bạn trong Amazon S3 bằng cách làm việc với đối tác từ Mạng lưới đối tác AWS (APN) — cộng đồng nhà cung cấp dịch vụ tư vấn và công nghệ lớn nhất. APN nhận diện các đối tác di chuyển truyền dữ liệu sang Amazon S3 và đối tác lưu trữ cung cấp giải pháp tích hợp S3 để phục vụ lưu trữ chính, sao lưu và khôi phục, lưu trữ và khôi phục sau thảm họa. Bạn cũng có thể mua giải pháp tích hợp AWS trực tiếp từ AWS Marketplace, tại đây có hơn 250 dịch vụ dành riêng cho lưu trữ.

## Trường hợp sử dụng Amazon S3

### Sao lưu và phục hồi

### Khôi phục sau thảm họa (DR)

### Lưu trữ

### Kho dữ liệu và phân tích dữ liệu lớn

### Lưu trữ đám mây lai

### Dữ liệu ứng dụng dành cho đám mây

**AWS RDS**

RDS là viết tắt của Amazon Relational Database Service là một web service cho phép bạn dễ dàng setup thao tác, scale cơ sở dữ liệu quan hệ trên AWS Cloud.

Amazon RDS sẽ đảm nhận các tác vụ khó hay các tác vụ quản lý:

* Bạn có thể phân bổ CPU, IOPS hay storage một cách tuỳ biến
* RDS sử dụng AWS backup service cho việc backup data, tự động phát hiện lỗi và recovery
* Không support việc access RDS instance thông qua shell
* Bạn có thể backup tự động hay thủ công Snapshot
* Khả năng tự đồng bộ cao giữa primary và secondary
* Kiểm soát được việc access vào RDS thông qua IAM, bảo vệ database bằng cách đẩy lên virtual private cloud

# **DB Instances**

DB instance là một kiểu môi trường cơ sở dữ liệu riêng biệt bên trong AWS cloud.

Bạn có thể tạo hay chỉnh sửa DB instance thông qua AWS CLI, RDS API và AWS management console

Mỗi một DB instance sẽ chạy một DB engine

RDS support các engine sau:

* MySQL
* MariaDB
* PostgreSQL
* Oraccle
* Microsoft SQL server

DB instance storage có 3 kiểu

* Magnetic
* General Purpose (SSD)
* Provisioned (PIOS) Mỗi một DB instace có giới hạn lưu trữ phụ thuộc vào kiểu và cơ sở dữ liệu mà nó support.

# **DB Instance Billing for Amazon RDS**

Việc tính toán tiền dựa vào các thành phần được sử dụng sau:

* DB instance hours (per hour): Dựa vào DB instance class của DB instance, Giá tiền sẽ được list ra theo giờ, Nhưng hoá đơn của bạn sẽ được tính toán theo giây và tối thiểu cho RDS billing là 10 phút
* Storage (GiB/tháng): Nếu bạn scale provisioned storage capacity trong tháng. Thì bill sẽ là pro-rated
* I/O request (1 triệu request mỗi tháng) : Áp dụng cho kiểu RDS magetic
* Provisioned IOPS (per IOPS per month)
* Backup storage (per GiB per month): Backup storage là việc lưu trữ được tự động liên kết với database backups
* Data transfer (per GB) : Là việc chuyển chuyển dữ liệu dữ instance với bên ngoài hoặc các service khác của AWS

# **Prerequirement**

Bạn sẽ tạo cơ sở dữ liệu bên trong DB instance.

DB instance sẽ cung cấp network address (endpoint). Ứng dụng sẽ sử dụng endpoint này để kết nối tới DB instance.

Bạn sẽ cần phải details setting storage, memory, engine / version, network, security, maintenance period khi tạo DB instance.

Kiểm soát việc truy cập DB instance thông qua security group

Trước khi tạo DB instance và security group bạn phải xác định được những thứ mà DB instance và security group cần Những điều quan trọng sau cần chú ý:

## **Resource requirements**

Memory và processor cần thiết cho ứng dụng

## **VPC, subnet và security group**

DB instance giống như VPC (Virtual Private Cloud). Để kêt nối được tới DB instance bạn cần cài đặt security goup rules. Những rules này là khác nhau đối với từng VPC mà bạn sử dụng

### **Default VPC**

Nếu AWS account của bạn có default VPC trên AWS region hiện tại, Thì VPC này đã được cấu hình để hỗ trợ cho DB instance. Nếu bạn chọn default VPC khi tạo DB instance thì thao tác theo bước sau:

* Tạo VPC security group để chứng thực kết từ ứng dùng tới Amazon RDS DB instance với database.
* Chọn default DB subnet group. Amazon RDS sẽ tạo default DB subnet group khi tạo DB instance nếu đây là lần đâu bạn tạo DB instance.

### **User-defined VPC**

Một số điều quan trọng cần chú ý

1. Hãy chắc chắn rằng VPC security group đã chứng thực việc kết nối giữa ứng dụng với từ Amazon RDS tới database.
2. VPC phải thoả mãn một số điều kiện để host DB instance như có ít nhất 2 subnets cho mỗi AZ
3. Sử dụng một DB subnet group để định nghĩa các subnets trong VPC

### **No VPC**

Nếu AWS account của bạn không có default VPC và bạn không chọn user-defined VPC

## **High availability**

Amazon RDS, Multi-AZ deployment sẽ tạo ra một primary DB instance và một secondary standby DB instance ở một AZ khác hỗ trợ cho việc chuyển đổi dự phòng khi có lỗi xảy ra.

## **IAM policies**

IAM account của bạn phải được phân quyền mà Amazon RDS yêu cầu.

## **Open ports**

Hãy đảm bảo firewall không block default port cho database engine. Bạn có thể sửa port bằng cách sửa DB instance

## **AWS Region**

Database và ứng dụng của bạn nên chọn region gần nhau để giảm độ trễ mạng

## **DB disk subsystem**

Tuỳ vào mục đích sử dụng mà bạn chọn 1 trong 3 loại Magnetic (Standard Storage), General Purpose (SSD), Provisioned IOPS (PIOPS)

**AWS EC2**

* ****Amazon Elastic Compute Cloud (Amazon EC2)**** là một cơ sở hạ tầng điện toán đám mây được cung cấp bởi ****Amazon Web Services (AWS)**** giúp cung cấp tài nguyên máy tính ảo hoá theo yêu cầu.
* ****Amazon EC2**** cung cấp các ứng dụng máy tính ảo hoá có thể mở rộng về khả năng xử lý cùng các thành phần phần cứng ảo như bộ nhớ máy tính (ram), vi xử lý, linh hoạt trong việc lựa chọn các phân vùng lưu trữ dữ liệu ở các nền tảng khác nhau và sự an toàn trong quản lý dịch vụ bởi kiến trúc ảo hoá đám mây mạnh mẽ của AWS.
* ****Amazon EC2**** sẽ cung cấp một hoặc máy chủ ảo có thể kết hợp với nhau để dễ dàng triển khai ứng dụng nhanh nhất và đảm bảo tính sẵn sàng cao nhất. Thậm chí về mặt thanh toán bạn dễ dàng biết được các mức chi phí cần thanh toán dựa trên thông tin tài nguyên bạn sử dụng.

### **Amazon EC2 Instance Là Gì?**

****Amazon EC2 Instanc**** là một cloud server. Với một tài khoản bạn có thể tạo và sử dụng nhiều ****Amazon EC2 Instance.**** Các Amazon EC2 Instance được chạy trên cùng một server vật lý và chia sẻ memory, CPU, ổ cứng...

Tuy nhiên do tính chất của cloud service nên mỗi một Instance được hoạt động giống như một server riêng lẻ.

### **Các đặc tính của Amazon EC2**

##### ****1. Scaling**:**

* Scaling Up/Down: Tăng/Giảm capacity(RAM, CPU,...) của Instance.
* Scaling In/Out: Tăng/Giảm số lượng Instance.

##### ****2. Security**:**

* Có thể thiết lập rank IP Private dành riêng cho EC2.
* Sử dụng Security Group và Network ACLS để control inbound/outbound.
* Có thể thiết lập IPsec VPN giữa Data Center và AWS Clound.
* Delicated Instance -> Tạo EC2 trên 1 hardware physical dành riêng cho 1 khách hàng duy nhất.

##### ****3. Cost**:**

* On-Demand Instance: Tính theo giờ, đáp ứng nhu cầu dùng trong thời gian ngắn. Dùng bao nhiêu, trả bấy nhiêu.
* Reserved Instance: Cho phép trả trước 1 lượng Server cho 1 hoặc 3 năm. Chi phí chỉ bằng 75% so với On-Demand. Nêú không có nhu cầu sử dụng trong thời gian dài, bạn có thể sale tại [Reserved Instance Marketplace](https://aws.amazon.com/ec2/purchasing-options/reserved-instances/marketplace/" \t "https://viblo.asia/p/_blank)