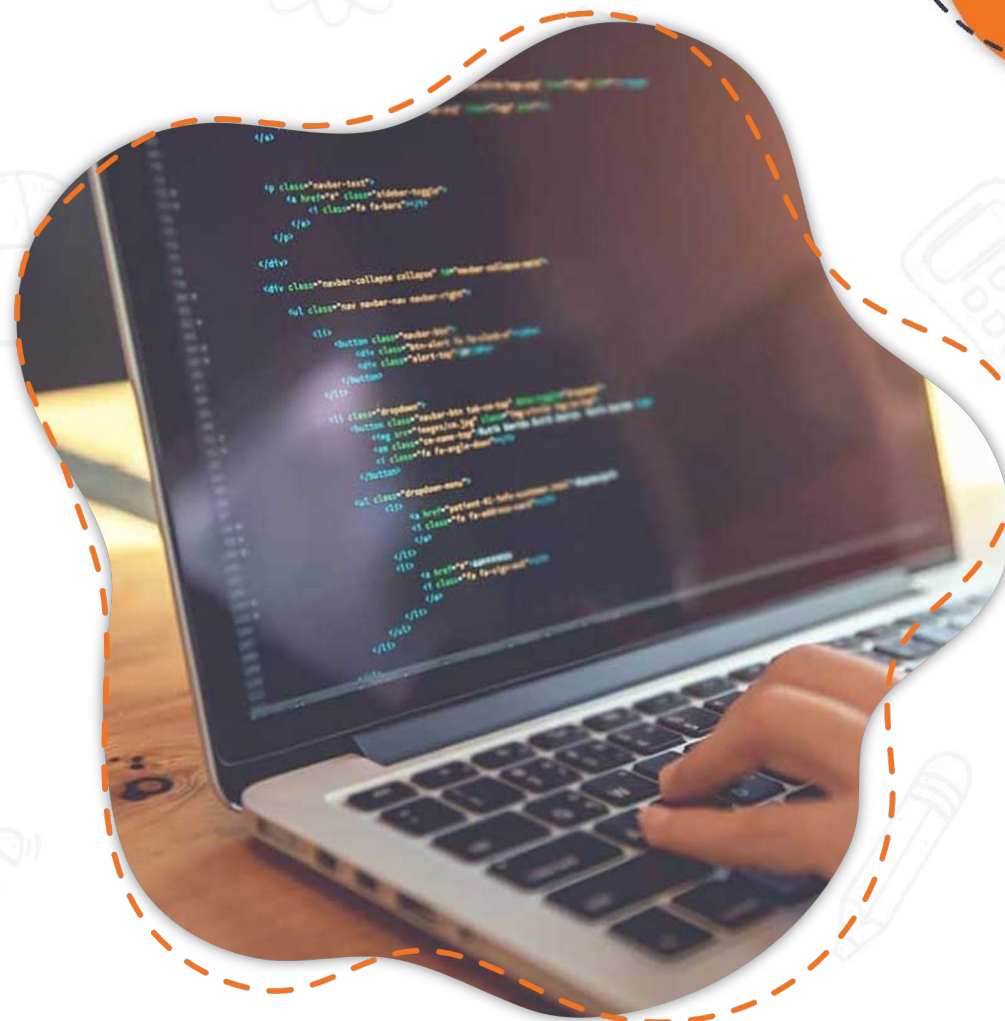


# C# Backend



**NỘI  
DUNG**

**01**

**Tổng quan ứng dụng ASP NET Core**

**02**

**Vòng đời của 1 request**

**03**

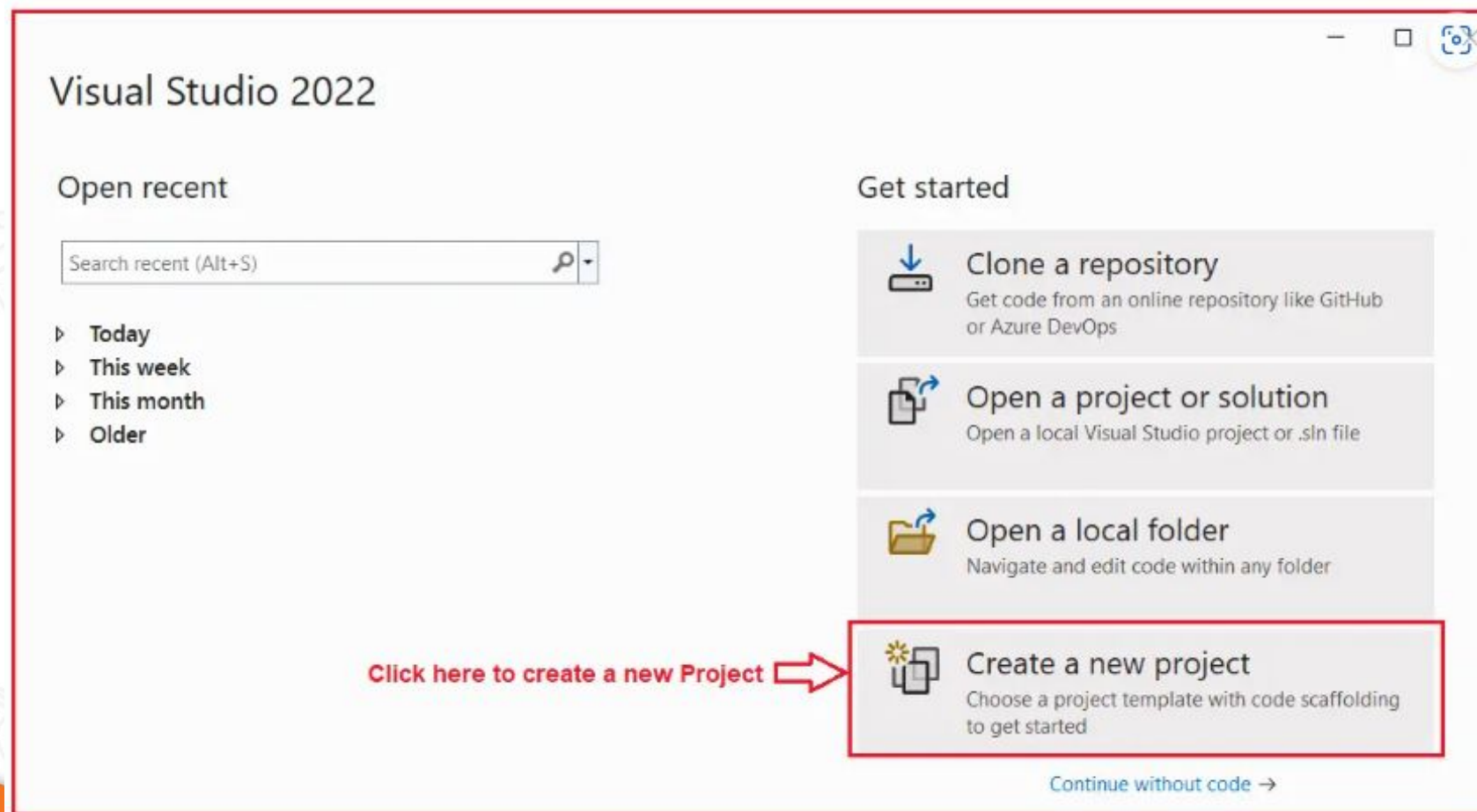
**Một số khái niệm cơ bản**

**04**

**Các môi trường trong ứng dụng thực tế**

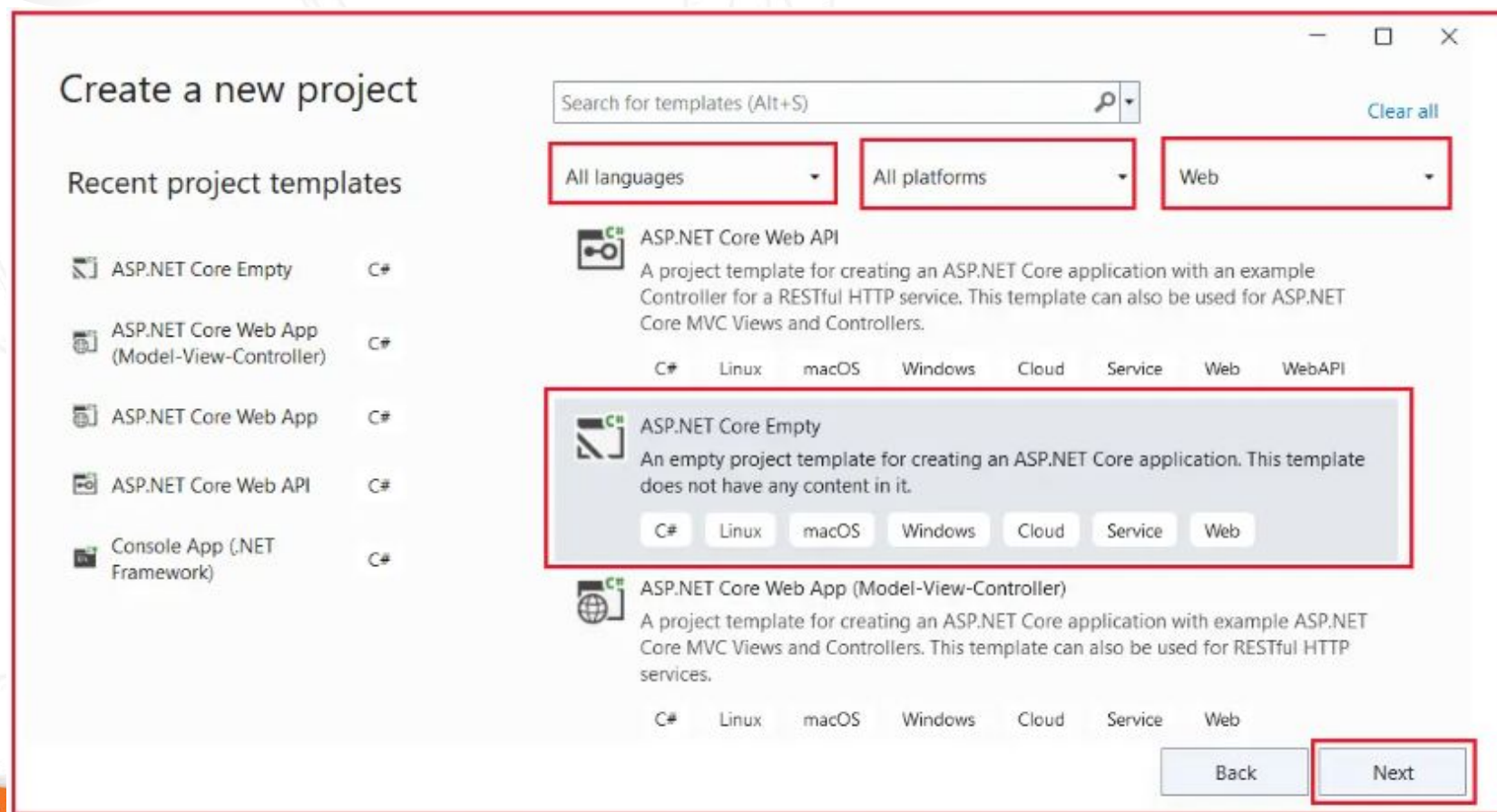
01

## Tổng quan ứng dụng ASP Net core



01

## Tổng quan ứng dụng ASP Net core





01

## Tổng quan ứng dụng ASP Net core

### Configure your new project

ASP.NET Core Empty C# Linux macOS Windows Cloud Service Web

Project name

FirstCoreWebApplication

Location

D:\Projects

Solution name ⓘ

FirstCoreWebApplication

☐ Place solution and project in the same directory

Project will be created in "D:\Projects\FirstCoreWebApplication\FirstCoreWebApplication\"

Back

Next

### Additional information

ASP.NET Core Empty C# Linux macOS Windows Cloud Service Web

Framework ⓘ

.NET 6.0 (Long Term Support)

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

Docker OS ⓘ

Linux

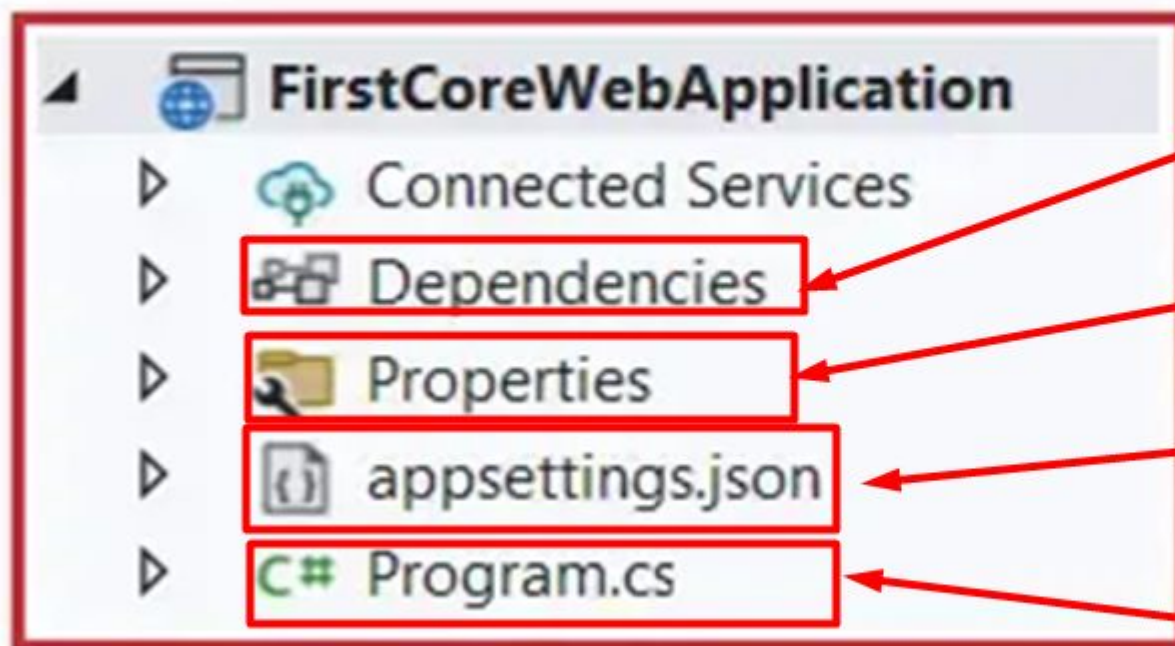
☒ Do not use top-level statements ⓘ

Back

Create

01

## Tổng quan ứng dụng ASP Net core



Contain nuget packages and referrenc projects

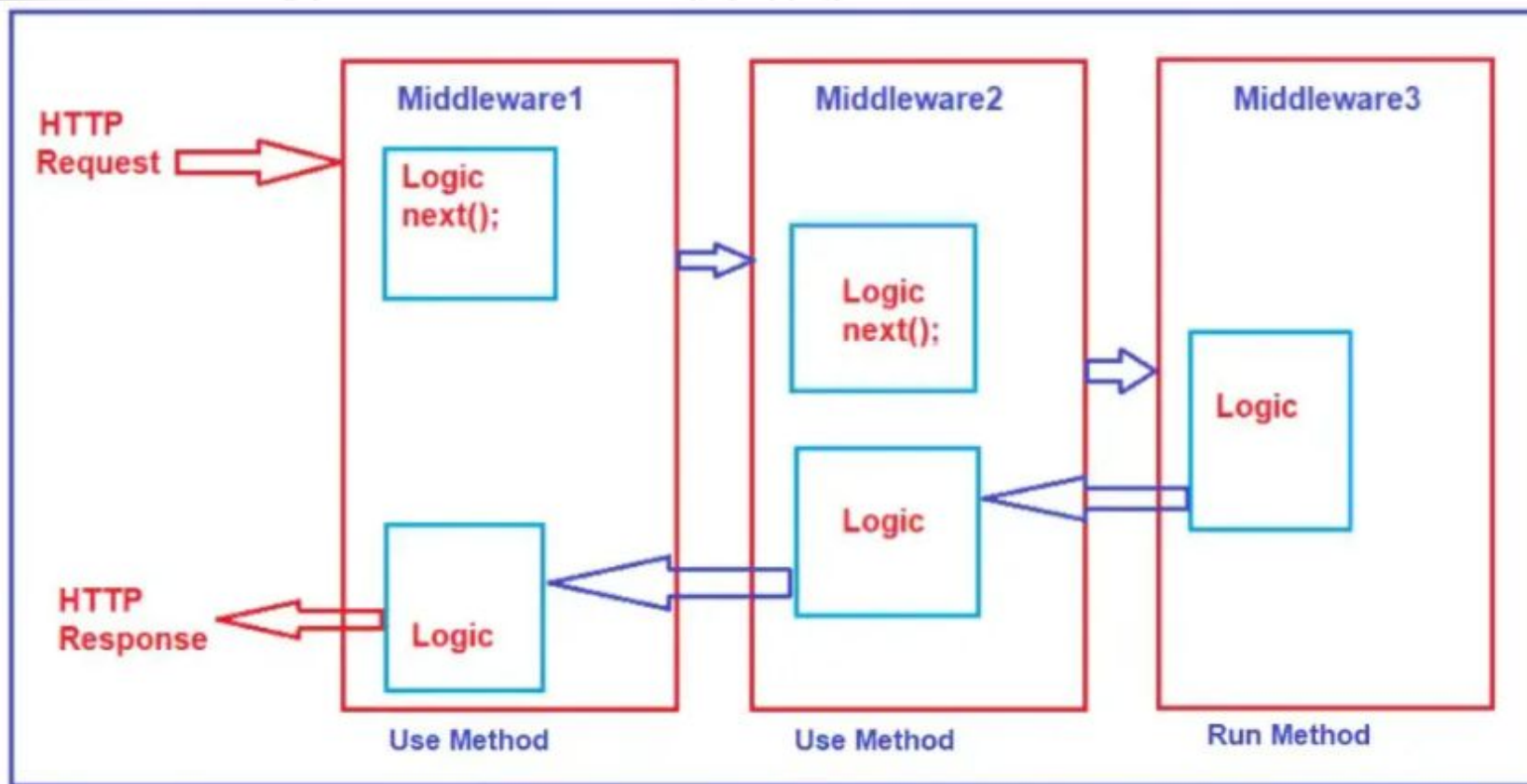
Contain profiles of the application

Contain configuration of the app  
ex: connectionstring, logging levels...

Run the web host

**01**

## Vòng đời của request trong ứng dụng



## 01

**Vòng đời của request trong ứng dụng**

### Request Pipeline

Request Pipeline là cơ chế bắt đầu khi các request bắt đầu được xử lý với một đối tượng Request đầu vào và kết thúc với đầu ra là một response. Pipeline chỉ ra cách mà ứng dụng phản hồi với HTTP Request. Request đến từ trình duyệt đi qua pipeline và quay trở lại khi xử lý xong để trả về client. Các thành phần đơn lẻ tạo nên pipeline này được gọi là middleware

### Middleware

Middleware là thành phần của phần mềm đóng vai trò tác động vào request pipeline (luồng request) để xử lý chúng và tạo ra response phản hồi lại client. Mỗi một tiến trình middleware thao tác với các request nhận được từ middleware trước nó. Nó cũng có thể quyết định gọi middleware tiếp theo trong pipeline hoặc trả về response cho middleware ngay trước nó. (ngắt pipeline).



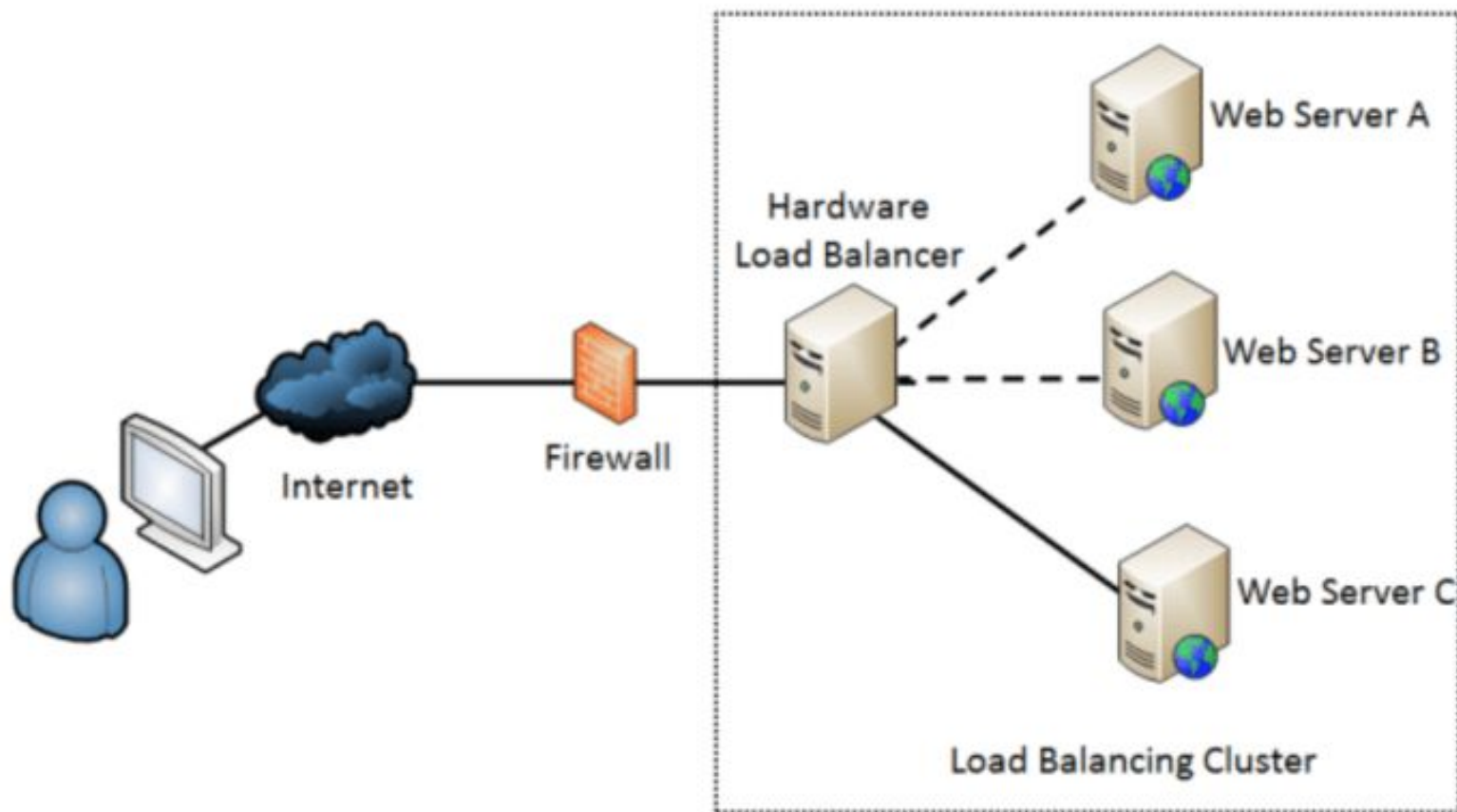
## 01

**Vòng đời của request trong ứng dụng**

- Đầu tiên, HTTP Request đến (trực tiếp hoặc qua External web server) ứng dụng. Kestrel web server đón lấy request và tạo một đối tượng **HttpContext** và gán nó vào Middleware đầu tiên trong request pipeline.
- Middleware đầu tiên sẽ nhận request, xử lý và gán nó cho middleware tiếp theo. Quá trình này tiếp diễn cho đến khi đi đến middleware cuối cùng. Tùy thuộc bạn muốn pipeline của bạn có bao nhiêu middleware.
- Middleware cuối cùng sẽ trả request ngược lại cho middleware trước đó, và sẽ ngắt quá trình trong request pipeline.
- Mỗi Middleware trong pipeline sẽ tuần tự có cơ hội thứ hai để kiểm tra lại request và điều chỉnh response trước khi được trả lại.
- Cuối cùng, response sẽ đến Kestrel nó sẽ trả response về cho client. Bất cứ middleware nào trong request pipeline đều có thể ngắt request pipeline tại chỗ đó với chỉ một bước đơn giản là không gán request đó đi tiếp.

01

## Một số khái niệm cơ bản



**01****Một số khái niệm cơ bản****Instance (Node)**

Một instance là một bản sao cụ thể của một ứng dụng hoặc dịch vụ đang chạy. Mỗi instance thường có một bản thể của mã và các tài nguyên của nó, được thực thi trong môi trường riêng biệt.

**Caching (Bộ nhớ cache):**

Caching là một kỹ thuật lưu trữ dữ liệu tạm thời để giảm thiểu thời gian truy cập dữ liệu từ nguồn gốc chính.

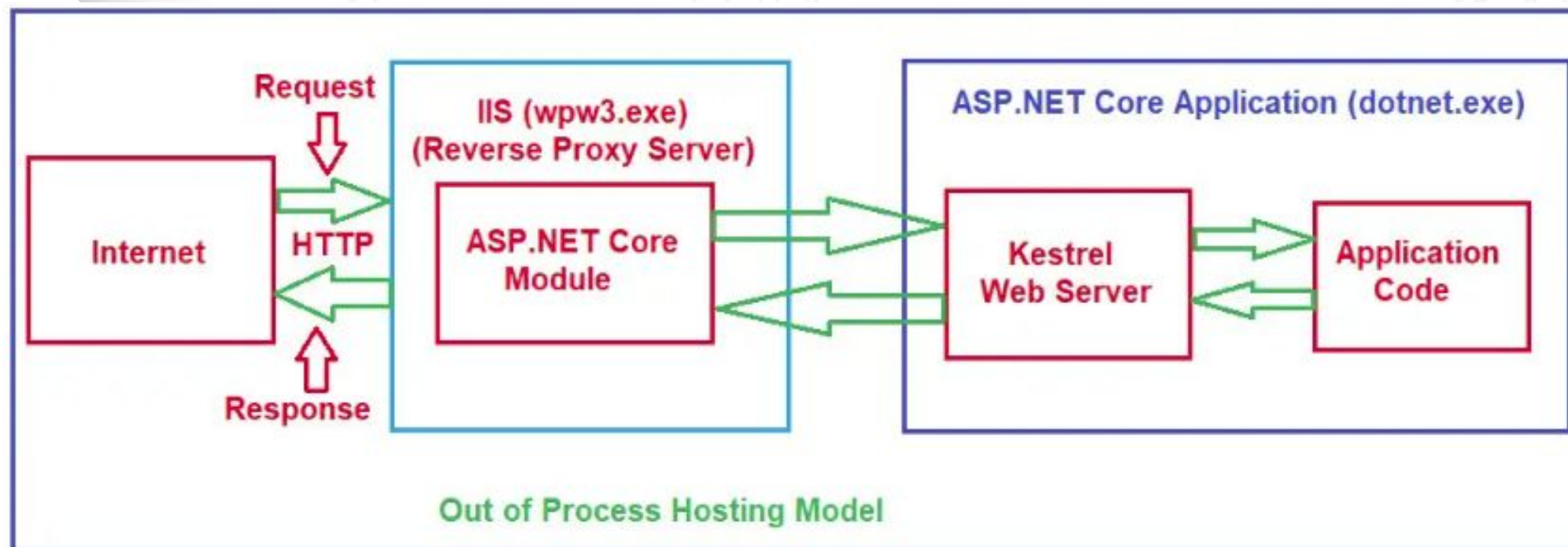
Khi dữ liệu được cache, các yêu cầu sau có thể được phục vụ nhanh hơn do không cần truy vấn lại nguồn dữ liệu chính.

**Load Balancer (Cân bằng tải)**

Load balancer là một thành phần trong kiến trúc hệ thống mạng được sử dụng để phân phối tải làm việc giữa nhiều máy chủ hoặc instance. Nó giúp cải thiện hiệu suất, độ tin cậy và khả năng mở rộng của hệ thống bằng cách phân chia công việc đều đặn và tự động giữa các máy chủ.

**01**

## Một số khái niệm cơ bản





## 01

## Một số khái niệm cơ bản

### **Kestrel Web Server**

- Kestrel là một web server được phát triển bởi Microsoft và được tích hợp sẵn trong ASP.NET Core.
- Nó là một web server gốc (native) và hiệu quả cao, được thiết kế để xử lý các yêu cầu HTTP và hỗ trợ cho việc xây dựng ứng dụng web ASP.NET Core. Nhờ đó mà ứng dụng net core có khả năng self host và cho phép chạy đa nền tảng

### **Nginx/Apache Web Server**

- Nginx và Apache là hai trong những web server phổ biến nhất trên thế giới, được sử dụng rộng rãi cho việc phục vụ các trang web và ứng dụng web. Hai loại web server này thường được cài đặt trên môi trường linux

### **IIS webserver**

- IIS là viết tắt của Internet Information Services, là một trong những web server phổ biến và mạnh mẽ nhất, được phát triển và hỗ trợ bởi Microsoft. IIS thường được cài đặt trên môi trường window

**01****Một số khái niệm cơ bản****Server Rendering**

Server rendering là quá trình tạo ra và gửi trang HTML hoàn chỉnh từ máy chủ đến trình duyệt của người dùng khi yêu cầu trang.

Trang web sử dụng server rendering thường là các trang web truyền thống, mà mỗi lần truy cập lại gửi yêu cầu đến máy chủ và máy chủ phản hồi bằng cách gửi trang HTML mới.

**Web API**

Trong trường hợp này, trang web thường chỉ chứa HTML cơ bản và các tệp tĩnh (ví dụ: CSS, JavaScript) và không có dữ liệu cụ thể.

Khi trang web cần dữ liệu hoặc thực hiện các thao tác nhất định, nó sẽ gửi các yêu cầu AJAX (Asynchronous JavaScript and XML) hoặc Fetch đến các API được xây dựng trên máy chủ hoặc bên ngoài.

API sau đó sẽ xử lý yêu cầu và trả về dữ liệu dưới dạng JSON, XML hoặc một định dạng dữ liệu khác mà trang web có thể hiểu và sử dụng.

## 01

**Một số khái niệm cơ bản****Môi trường QA (Quality Assurance)**

Môi trường QA là nơi mà các bản phát triển mới của phần mềm được kiểm tra và đánh giá.

Nhóm QA thường tiến hành kiểm thử tự động và thủ công để xác định lỗi và đảm bảo chất lượng của phần mềm trước khi nó được triển khai vào môi trường staging

**Môi trường Staging**

Các bản vá lỗi và tính năng mới được triển khai trong môi trường staging để kiểm tra tích hợp và kiểm tra hệ thống.

Môi trường staging cũng cung cấp cho các bên liên quan (bao gồm cả nhóm phát triển và khách hàng) cơ hội kiểm tra và xác nhận rằng các tính năng mới hoạt động đúng như mong đợi.

**Môi trường Production**

Môi trường production là môi trường thực tế mà người dùng cuối sử dụng phần mềm.

Các bản phát hành đã được kiểm tra kỹ lưỡng trong môi trường QA và staging trước khi được triển khai vào môi trường production.