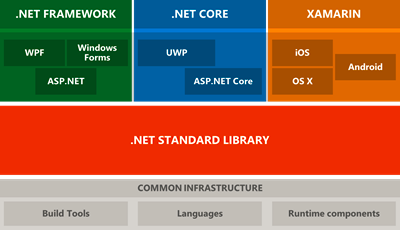
一、现有的.net生态圈：

****

.NET STANDARD LIBRARY是.net 标准库，.NET STANDARD是一套正式的.NET API规范，任何基于.NET STANDARD开发的类库，都能在上面提到的三个运行时中运行，也就是说上面的三类程序都可以引用.NET STANDARD LIBRARY。

.NET FRAMEWORK、.NET CORE和XAMARIN是三个不同的运行时，这三个运行时在.NET STANDARD LIBRARY之上，它们都包含一些基础的组件，像编译器、支持的开发语言（C#、F#等）、运行时组件（类型系统、垃圾回收等）。

这三个运行时是同一类的东西，只不过侧重点不同：

* **XAMARIN**：主要用来构建APP（IOS、Android、Windows）；
* **.NET Framework:** 这个是我们现在经常用的，可以创建ASP.NET Web 窗体应用程序、windows应用程序还有WCF服务器，wpf 等等相关的应用程序，web 方面就是Asp.net MVC；
* **.NET Core：**是微软推出的最新的开源的跨平台的框架，用它可以创建的应用可以运行在MAC，Linux上，.net core 支持UWP和 ASP.NET Core；

UWP即[Windows 10](http://baike.baidu.com/view/1110471.htm" \t "_blank) 中的Universal Windows Platform简称。即Windows通用应用平台，在Win 10 Mobile/Surface(Windows平板电脑）/[PC](http://baike.baidu.com/subview/13621/4882082.htm)/[Xbox](http://baike.baidu.com/subview/13669/20320565.htm)/[HoloLens](http://baike.baidu.com/item/HoloLens" \t "_blank)等平台上运行，uwp不同于传统pc上的exe应用也跟只适用于手机端的app有本质区别。它并不是为某一个终端而设计，而是可以在所有windows10设备上运行。

二、.net core与.net framework使用情况：

在以下情况，对服务器应用程序使用 .NET Core：

用户有跨平台需求。

用户正在面向微服务。

用户正在使用 Docker 容器。

需要高性能和可扩展的系统。

需按应用程序提供并行的 .NET 版本。

在以下情况，对服务器应用程序使用 .NET Framework ：

应用当前使用 .NET Framework（建议扩展而不是迁移）。

应用使用不可用于 .NET Core 的第三方 .NET 库或 NuGet 包。

应用使用不可用于 .NET Core 的 .NET 技术。

应用使用不支持 .NET Core 的平台。

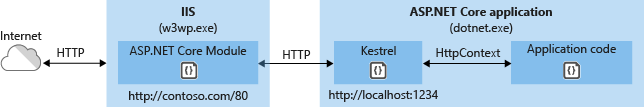
三、asp.net core概念

Asp.net Core不是 ASP.NET 的升级，而是一个重新设计的Web开发框架；一个ASP.NET CORE应用其实就是一个在其Main方法中创建一个Web服务器的简单控制台应用程序。

四、asp.net core工作原理：

ASP.NET Core 托管模式并不直接监听请求；而是依赖于一个 HTTP server 实现来转发请求到应用程序。这个被转发的请求会以一组 feature 接口的形式被包装，然后被应用程序组合到一个 HttpContext中去。 ASP.NET Core 包含了一个托管的跨平台 web 服务器，被称为 Kestrel，它往往会被运行在一个如 IIS 或者 nginx 的生产 web 服务器中。

ASP.NET Core 包含一个叫做 Kestrel 的 Web 服务器，基于libuv（一个跨平台的异步I/O library），主要用于 iOS 和 Linux 操作系统。Kestrel是一个跨平台的web服务器，与IIS一样负责请求的监听、接收和响应，但没有IIS丰富的管理功能。



IIS负责把外部请求转发给后端的Asp.net Core App，充当“反向代理（reverse-proxy）”角色。

“反向”是指入站方向，“代理”可以理解为某种服务，这里是转发HTTP请求的服务。“反向代理”就是入站方向转发HTTP请求的服务。

ASP.NET Core中可以用两种HTTP server，它们请求原理如图所示：

Kestrel 是跨平台的HTTP server 基于libuv（一个跨平台的异步I/O library）

Httpsys 是基于Windows内核驱动程序Http.Sys的HTTP server。Http.Sys是成熟的技术，可以防范多种攻击，并提供全功能Web服务器的鲁棒性，安全性和可扩展性。IIS本身作为HTTP侦听器运行在Http.Sys之上。

在 ASP.NET Core 中，IIS 是通过 HTTP 的方式来调用我们的 ASP.NET Core 程序。而部署在IIS中时，并不需要我们手动来启动 ASP.NET Core 的控制台程序，这是因为IIS新增了一个 AspNetCoreModule 模块，它负责 ASP.NET Core 程序的启动与停止，并能监听 ASP.NET Core 程序的状态，在我们的应用程序意外崩溃时重新启动。