**.NET Framework vs .NET Core: Key Differences**

.NET Framework and .NET Core are both software development platforms created by Microsoft, but they serve different purposes and have distinct characteristics.

**1. Overview**

**.NET Framework**

* Introduced in **2002**, primarily for Windows applications.
* A **mature, full-featured** framework used for enterprise applications.
* Supports **Windows-only** development.
* Uses **CLR (Common Language Runtime)** for execution.

**.NET Core**

* Introduced in **2016** as a **cross-platform, open-source** alternative.
* Designed for **modern, high-performance, and cloud-based applications**.
* Runs on **Windows, macOS, and Linux**.
* Uses **CoreCLR**, a lightweight and modular runtime.

**2. Key Differences**

| **Feature** | **.NET Framework** | **.NET Core** |
| --- | --- | --- |
| **Platform** | Windows-only | Cross-platform (Windows, Linux, macOS) |
| **Performance** | Slower due to overhead | Faster and optimized for performance |
| **Modularization** | Monolithic, large libraries | Lightweight, modular (NuGet packages) |
| **Deployment** | Requires full framework installed | Can be deployed as self-contained or framework-dependent |
| **Application Types** | Windows Forms, WPF, ASP.NET Web Forms, ASP.NET MVC | ASP.NET Core, Blazor, Console Apps, Microservices |
| **Open Source** | Partially open-source | Fully open-source (on GitHub) |
| **Microservices Support** | Limited | Optimized for microservices and containers |
| **Future Development** | No further major updates | Actively developed and evolving |

**3. When to Use?**

✅ **Use .NET Framework if**:

* You need **Windows-only** applications.
* You rely on **legacy** technologies like **WPF, Windows Forms, or Web Forms**.
* You have a large existing **enterprise** application built on .NET Framework.

✅ **Use .NET Core if**:

* You need a **cross-platform** application.
* You want **high performance and scalability**.
* You're developing **microservices, cloud-based apps, or containerized applications**.

**4. The Future: .NET 5+ and Beyond**

Microsoft **merged .NET Framework and .NET Core into a unified platform** called **.NET 5 and later versions (.NET 6, 7, 8, etc.)**. The future of .NET is based on **.NET Core**, with **.NET Framework receiving only maintenance updates**.