Historical Evolution of .NET Framework → .NET Core → Modern .NET

Evolution .NET Framework 4.5 → 4.8.1

NET Framework 4.5 (2012)

- Introduced async/await (C# 5)
- Improved ThreadPool scalability
- Modern HttpClient introduced
- Improved Background GC

.NET Framework 4.5.1 / 4.5.2

- Edit-and-continue improvements
- Better ASP.NET performance
- High DPI improvements
- Better exception diagnostics

.NET Framework 4.6

- RyuJIT 64-bit JIT compiler introduced
- TLS 1.2 support
- Roslyn compiler enables C# 6

.NET Framework 4.6.1 / 4.6.2

- Improved cryptography & X509
- Always Encrypted for SQL Server
- Compression performance improvements

.NET Framework 4.7 / 4.7.1 / 4.7.2

- High-DPI improvements
- .NET Standard 2.0 compatibility
- GC performance improvements

.NET Framework 4.8 / 4.8.1

- RyuJIT upgrades
- LOH (Large Object Heap) compacting improvements
- Accessibility improvements
- ARM64 support

Modern Runtime Concepts (Roslyn, JIT, PGO, Native AOT, GC)

Roslyn Compiler

Roslyn replaced the old proprietary compiler with an open-source, fully managed compiler. It provides APIs for code analysis, refactoring, and live IDE feedback.

RyuJIT & Modern JITing

RyuJIT is the modern Just-In-Time compiler delivering faster throughput and better SIMD support.

Dynamic PGO (Profile-Guided Optimization)

The runtime collects execution data and optimizes methods on the fly, improving hot-path performance.

Native AOT

Ahead-of-Time compilation produces a fully native executable, reducing startup time and memory footprint.

Advanced Garbage Collection

Modern .NET GC reduces latency, improves fragmentation control, and adds LOH compacting.

RefreshMemoryLimit

Feature for containerized environments that lets .NET pause or compact memory before OOM, improving stability.