

Hospedar e implementar ASP.NET Core

En general, para implementar una aplicación ASP.NET Core en un entorno de alojamiento:

- Implemente la aplicación publicada en una carpeta en el servidor de alojamiento.
- Configure un administrador de procesos que inicie la aplicación cuando lleguen las solicitudes y reinicie la aplicación después de que falle o el servidor se reinicie.
- Para la configuración de un proxy inverso, configure un proxy inverso para reenviar solicitudes a la aplicación.

Publicar en una carpeta

El comando de [publicación dotnet](#) compila el código de la aplicación y copia los archivos necesarios para ejecutar la aplicación en una carpeta de *publicación* . Al implementar desde Visual Studio, el proceso `dotnet publish` se produce automáticamente antes de que los archivos se copien en el destino de implementación.

Contenido de la carpeta

La carpeta de *publicación* contiene uno o más archivos de ensamblaje de aplicaciones, dependencias y, opcionalmente, el tiempo de ejecución de .NET.

Una aplicación .NET Core puede publicarse como *implementación independiente* o *implementación dependiente del FRAMEWORK*. Si la aplicación es autónoma, los archivos de ensamblaje que contienen el tiempo de ejecución .NET se incluyen en la carpeta de *publicación* . Si la aplicación depende del framework, los archivos de tiempo de ejecución de .NET no se incluyen porque la aplicación tiene una referencia a una versión de .NET que está instalada en el servidor. El modelo de implementación predeterminado depende del framework. Para obtener más información, vea [Implementación de la aplicación .NET Core](#) .

Además de los archivos `.exe` y `.dll` , la carpeta de *publicación* para una aplicación ASP.NET Core generalmente contiene archivos de configuración, activos estáticos y vistas MVC. Para obtener más información, consulte [Estructura de directorio de ASP.NET Core](#) .

Configurar un administrador de procesos

Una aplicación ASP.NET Core es una aplicación de consola que debe iniciarse cuando se inicia un servidor y reiniciarse si falla. Para automatizar los inicios y reinicios, se requiere un administrador de procesos. Los administradores de procesos más comunes para ASP.NET Core son:

- Linux
 - [Nginx](#)
 - [apache](#)
- windows
 - [IIS](#)
 - [Windows](#) Service

Configurar un proxy inverso

Si la aplicación usa el servidor [Kestrel](#) , [Nginx](#) , [Apache](#) o [IIS](#) pueden usarse como un servidor proxy inverso. Un servidor proxy inverso recibe solicitudes HTTP de Internet y las reenvía a Kestrel.

Cualquiera de las configuraciones, con o sin un servidor proxy inverso, es una configuración de alojamiento compatible. Para obtener más información, consulte [Cuándo usar Kestrel con un proxy inverso](#) .

Servidor proxy y escenarios de equilibrador de carga

Es posible que se requiera una configuración adicional para aplicaciones alojadas detrás de servidores proxy y equilibradores de carga. Sin una configuración adicional, una aplicación podría no tener acceso al esquema (HTTP / HTTPS) y la dirección IP remota donde se originó una solicitud. Para obtener más información, consulte [Configurar ASP.NET Core para trabajar con servidores proxy y equilibradores de carga](#) .

Use Visual Studio y MSBuild para automatizar implementaciones

La implementación a menudo requiere tareas adicionales además de copiar el resultado de la [publicación dotnet](#) a un servidor. Por ejemplo, se pueden requerir o excluir archivos adicionales de la carpeta de *publicación* . Visual Studio usa MSBuild para la implementación web, y MSBuild se puede personalizar para realizar muchas otras tareas durante la implementación. Para

obtener más información, consulte los [perfiles de publicación de Visual Studio \(.pubxml\) para la implementación de la aplicación ASP.NET Core](#) y el libro [Uso de MSBuild y Team Foundation Build](#) .

Mediante el uso de [la función Publicar Web](#) o la [compatibilidad integrada con Git](#) , las aplicaciones se pueden implementar directamente desde Visual Studio en el Servicio de aplicaciones de Azure. Azure DevOps Services admite [la implementación continua en Azure App Service](#) . Para obtener más información, vea [DevOps con ASP.NET Core y Azure](#) .

Publicar en Azure

Consulte [Publicar una aplicación ASP.NET Core en Azure con Visual Studio](#) para obtener instrucciones sobre cómo publicar una aplicación en Azure con Visual Studio. Un ejemplo adicional lo proporciona [Crear una aplicación web ASP.NET Core en Azure](#) .

Publicar con MSDeploy en Windows

Consulte los [perfiles de publicación de Visual Studio \(.pubxml\) para la implementación de la aplicación ASP.NET Core](#) para obtener instrucciones sobre cómo publicar una aplicación con un perfil de publicación de Visual Studio, incluso desde un símbolo del sistema de Windows utilizando el comando [dotnet msbuild](#) .

Servicios de información de Internet (IIS)

Para implementaciones en Internet Information Services (IIS) con la configuración proporcionada por el archivo *web.config* , consulte los artículos en [Host ASP.NET Core en Windows con IIS](#) .

Alojar en web farm

Para obtener información sobre la configuración para alojar aplicaciones ASP.NET Core en un entorno de servidores web farm (por ejemplo, implementación de varias instancias de su aplicación para la escalabilidad), consulte [Host ASP.NET Core en de servidores farm web](#) .

Anfitrión en Docker

Para obtener más información, consulte [Host ASP.NET Core en contenedores Docker](#) .

Realizar controles de estado

Use Health Check Middleware para realizar comprobaciones de estado en una aplicación y sus dependencias. Para obtener más información, consulte [Verificaciones de estado en ASP.NET Core](#) .

Recursos adicionales

- [Solucionar problemas y depurar proyectos de ASP.NET Core](#)
- [Hospedaje ASP.NET](#)