# **Hospedaje de una aplicación web con Azure App Service**

* 31 min
* Módulo
* 8 Unidades

4.6 (6,871)

Calificarlo

PrincipianteDesarrolladorArquitecto de solucionesEstudianteAzureApp ServiceAzure Portal

Azure App Service le permite crear y hospedar aplicaciones web en el lenguaje de programación que prefiera sin tener que administrar la infraestructura. Aprenda a crear un sitio web mediante la plataforma de aplicaciones web hospedada en Azure App Service.

## **Objetivos de aprendizaje**

En este módulo, aprenderá a:

* Usar Azure Portal para crear una aplicación web de Azure App Service.
* Usar herramientas de desarrollo para crear el código para una aplicación web de inicio.
* Implementar el código en Azure App Service.

# **Introducción**

Completado

100 XP

* 2 minutos

Imagine que está compilando un sitio web para una nueva empresa o que está ejecutando una aplicación web existente en un servidor local antiguo. Configurar un nuevo servidor puede representar todo un desafío. Es necesario disponer del hardware adecuado, preferentemente un sistema operativo de nivel de servidor, y una pila de hospedaje web.

Una vez que esté en ejecución, debe mantener el servidor. Y, ¿qué ocurre si aumenta el tráfico del sitio web? Es posible que deba invertir en hardware adicional.

Hospedar la aplicación web con Azure App Service hace que la implementación y administración de una aplicación web resulten mucho más fáciles en comparación con la administración de un servidor físico. En este módulo, se implementará una aplicación web en Azure App Service.

## **Objetivos de aprendizaje**

En este módulo, aprenderá a:

* Usar Azure Portal para crear una aplicación web de Azure App Service.
* Usar herramientas de desarrollo para crear el código para una aplicación web de inicio.
* Implementar el código en App Service

# **Creación de una aplicación web en Azure Portal**

Completado

100 XP

* 6 minutos

En esta unidad, aprenderá a crear una aplicación web de Azure App Service con Azure Portal.

## **Por qué usar Azure Portal**

El primer paso para hospedar la aplicación web consiste en crear una aplicación web (aplicación de App Service) dentro de la suscripción de Azure.

Hay varias maneras de crear una aplicación web. Puede usar Azure Portal, la interfaz de línea de comandos (CLI) de Azure, un script o un IDE.

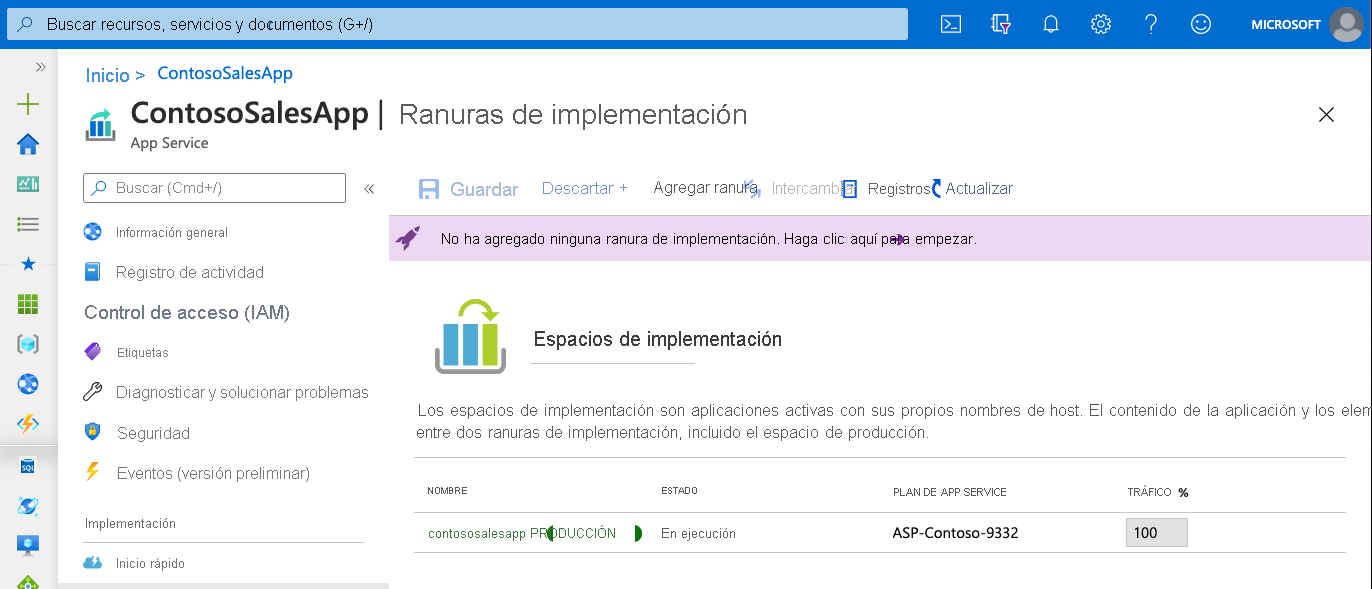
En la información que se presenta a continuación se describe cómo usar Azure Portal para crear una aplicación web y, en el siguiente ejercicio, usará esta información para crearla. En este módulo, la demostración se realizará con Azure Portal porque es una experiencia gráfica, lo que lo convierte en una excelente herramienta de aprendizaje. El portal ayuda a descubrir las características disponibles, agregar recursos adicionales y personalizar los recursos existentes.

## **Qué es Azure App Service**

Azure App Service es una plataforma de hospedaje de aplicaciones web totalmente administrada. Esta plataforma como servicio (PaaS) que ofrece Azure le permite centrarse en el diseño y la compilación de la aplicación, mientras que Azure se encarga de la infraestructura para ejecutar y escalar las aplicaciones.

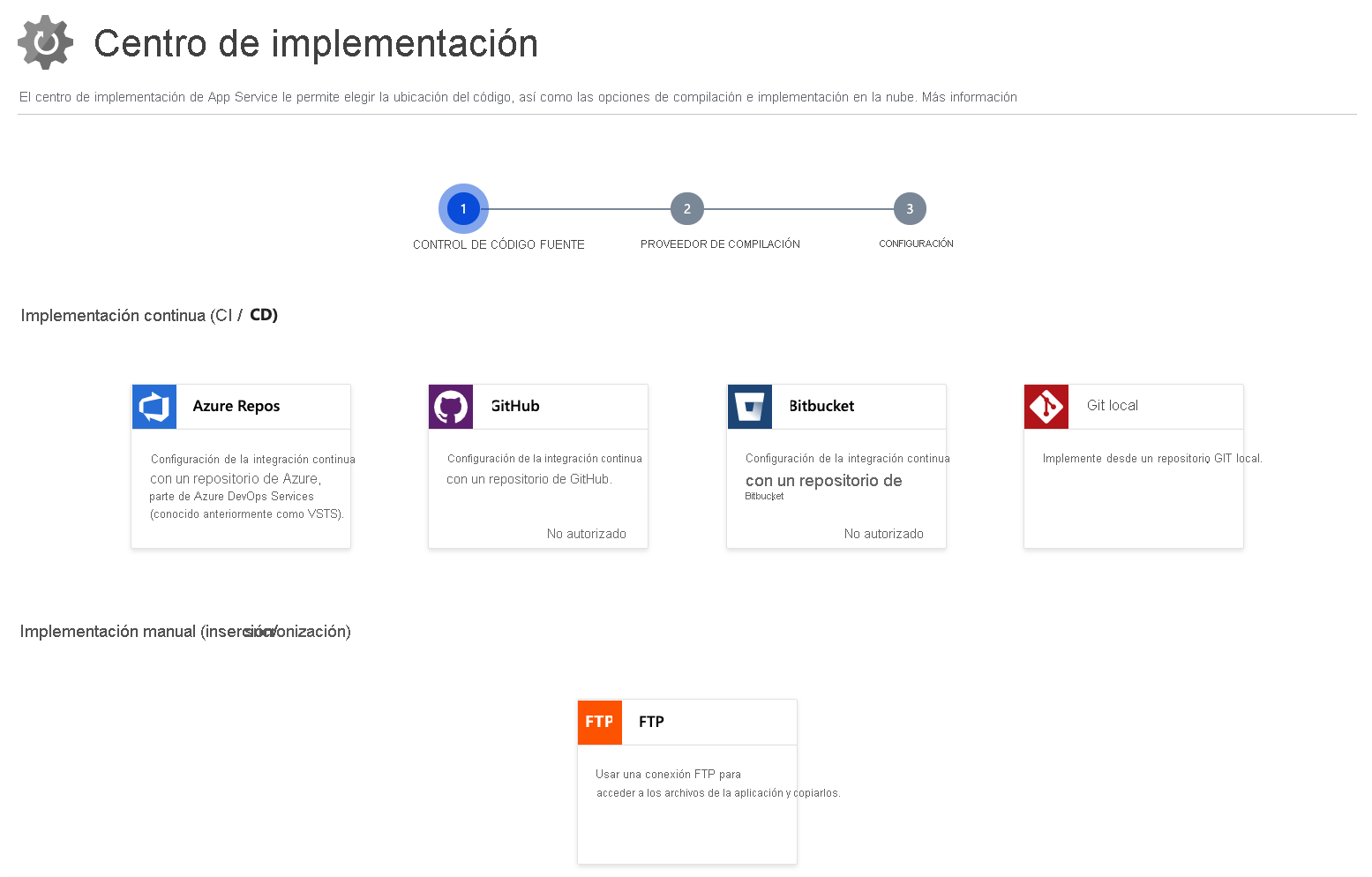
### **Ranuras de implementación**

Con Azure Portal, puede agregar fácilmente ranuras de implementación a una aplicación web de App Service. Por ejemplo, puede crear un espacio de implementación de ensayo donde puede insertar el código para probarlo en Azure. Cuando esté satisfecho con el código, le resultará muy fácil intercambiar la ranura de implementación de ensayo con la ranura de producción. Todo esto se hace con unos pocos clics del ratón en Azure Portal.



### **Compatibilidad con la integración e implementación continuas**

Azure Portal proporciona integración e implementación continuas listas para usar con Azure DevOps, GitHub, Bitbucket, FTP o un repositorio de GIT local en el equipo de desarrollo. Conecte la aplicación web con cualquiera de los orígenes anteriores y App Service se encargará del resto mediante la sincronización automática del código y los futuros cambios del código en la aplicación web. Además, con Azure DevOps, puede definir un proceso propio de compilación y publicación que compile el código fuente, ejecute las pruebas, cree una versión y, por último, la implemente en la aplicación web cada vez que confirme el código. Todo eso tiene lugar de forma implícita sin ninguna intervención.



### **Publicación de Visual Studio integrada y publicación FTP**

Además de poder configurar la integración e implementación continuas de la aplicación web, siempre se puede beneficiar de la estrecha integración con Visual Studio para publicar la aplicación web en Azure mediante la tecnología Web Deploy. App Service también admite la publicación basada en FTP para flujos de trabajo más tradicionales.

### **Compatibilidad integrada con la escalabilidad automática (escalabilidad horizontal en función de la carga real)**

En la aplicación web se integra la capacidad de escalar o reducir verticalmente, o bien escalar horizontalmente. En función del uso de la aplicación web, se puede escalar o reducir verticalmente si se aumentan o disminuyen los recursos de la máquina subyacente en la que se hospeda la aplicación web. Los recursos pueden ser el número de núcleos o la cantidad de memoria RAM disponible.

Por otro lado, el escalado horizontal es la capacidad de aumentar el número de instancias de máquina que ejecutan la aplicación web.

## **Creación de una aplicación web**

Cuando esté listo para ejecutar una aplicación web en Azure, visite Azure Portal y cree un recurso de aplicación web. Al crear una aplicación web se asigna un conjunto de recursos de hospedaje en App Service, que se puede usar para hospedar cualquier aplicación basada en web compatible con Azure, ya sea en ASP.NET Core, Node.js, Java, Python, etc.

En Azure Portal se proporciona un asistente para crear una aplicación web. Este asistente requiere los campos siguientes:

| ***Campo*** | ***Descripción*** |
| --- | --- |
| Suscripción | una suscripción a Azure válida y activa. |
| Grupo de recursos | un grupo de recursos válido. |
| Nombre de la aplicación | el nombre de la aplicación web. Este nombre se convierte en parte de la dirección URL de la aplicación, por lo que debe ser único entre todas las aplicaciones web de Azure App Service. |
| Publicar | puede implementar la aplicación en App Service como código o como una imagen de Docker lista para ejecutarse. Al seleccionar Imagen de Docker se activará la pestaña Docker del asistente, donde se proporciona información sobre el registro de Docker desde el que App Service recuperará la imagen. |
| Pila del entorno en tiempo de ejecución | si elige implementar la aplicación como código, App Service necesita saber qué entorno de ejecución usa la aplicación (entre los ejemplos se incluyen Node.js, Python, Java y .NET). Si implementa la aplicación como una imagen de Docker, no tendrá que elegir una pila del entorno en tiempo de ejecución, ya que la imagen la incluirá. |
| Sistema operativo | App Service puede hospedar aplicaciones en servidores Windows o Linux. Para más información, consulte a continuación. |
| Región | la región de Azure desde la que se distribuirá la aplicación. |
| Plan de App Service | para más información sobre los planes de App Service, consulte a continuación. |

### **Sistemas operativos**

Si va a implementar la aplicación como código, muchas de las pilas del entorno en tiempo de ejecución disponibles están limitadas a un sistema operativo o a otro. Después de elegir una pila en tiempo de ejecución, el botón de alternancia indicará si puede elegir un sistema operativo. Si la pila en tiempo de ejecución de destino está disponible en ambos sistemas operativos, seleccione el que use para desarrollar y probar la aplicación.

Si la aplicación está empaquetada como una imagen de Docker, elija el sistema operativo en el que se ha diseñado la imagen para ejecutarse.

Al seleccionar Windows, se activa la pestaña Supervisión, donde tiene la opción de habilitar Application Insights. Al habilitar esta característica, se configurará la aplicación para que envíe de forma automática telemetría de rendimiento detallada al servicio de supervisión de Application Insights sin necesidad de realizar cambios en el código. Application Insights también se puede usar desde aplicaciones hospedadas en Linux, pero esta opción inmediata sin código solo está disponible en Windows.

### **Planes de App Service**

Un plan de App Service es un conjunto de recursos de servidor virtual que ejecutan aplicaciones de App Service. El tamaño de un plan (que a veces se denomina SKU o plan de tarifa) determina las características de rendimiento de los servidores virtuales que ejecutan las aplicaciones asignadas al plan y las características de App Service a las que tienen acceso estas aplicaciones. Todas las aplicaciones web de App Service que cree deben asignarse a un único plan de App Service que las ejecute.

Un solo plan de App Service puede hospedar varias aplicaciones web de App Service. En la mayoría de los casos, el número de aplicaciones que se pueden ejecutar en un solo plan estará limitado por las características de rendimiento de las aplicaciones y las limitaciones de recursos del plan.

Los planes de App Service son la unidad de facturación de App Service. El tamaño de cada plan de App Service de la suscripción, además de los recursos de ancho de banda que usan las aplicaciones implementadas en esos planes, determina el precio que se paga. El número de aplicaciones web implementadas en los planes de App Service no tiene ningún efecto en la factura.

Puede usar cualquiera de las herramientas de administración de Azure disponibles para crear un plan de App Service. Al crear una aplicación web a través de Azure Portal, el asistente le ayudará a crear un plan al mismo tiempo si todavía no tiene uno.

# **Ejercicio: Creación de una aplicación web en Azure Portal**

600 XP

* 5 minutos

Este módulo requiere un espacio aislado para completarse. Usó 1 de 10 espacios aislados por hoy. Mañana habrá disponibles más espacios aislados.

Activar espacio aislado

Elija el lenguaje de desarrollo

C#

Java

Node.js

Python

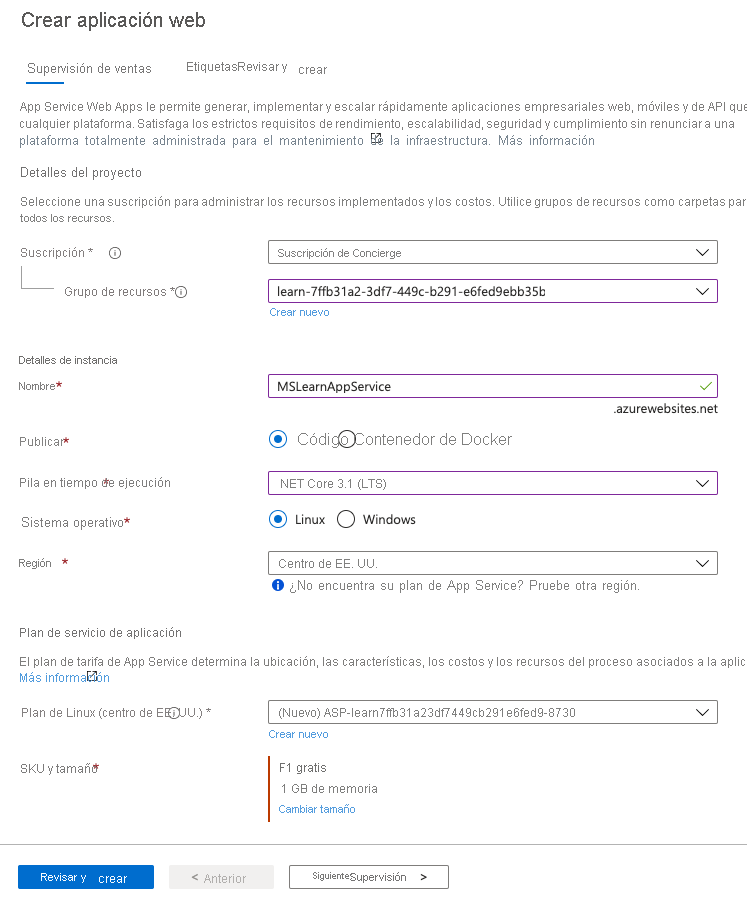
En esta unidad, se usará Azure Portal para crear una aplicación web.

## **Creación de una aplicación web**

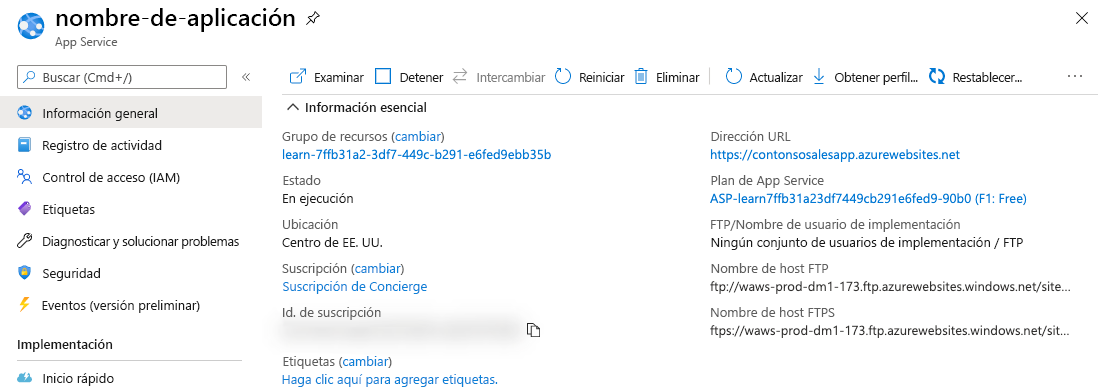
Inicie sesión en [Azure Portal](https://portal.azure.com/learn.docs.microsoft.com) con la misma cuenta con la que ha activado el espacio aislado.

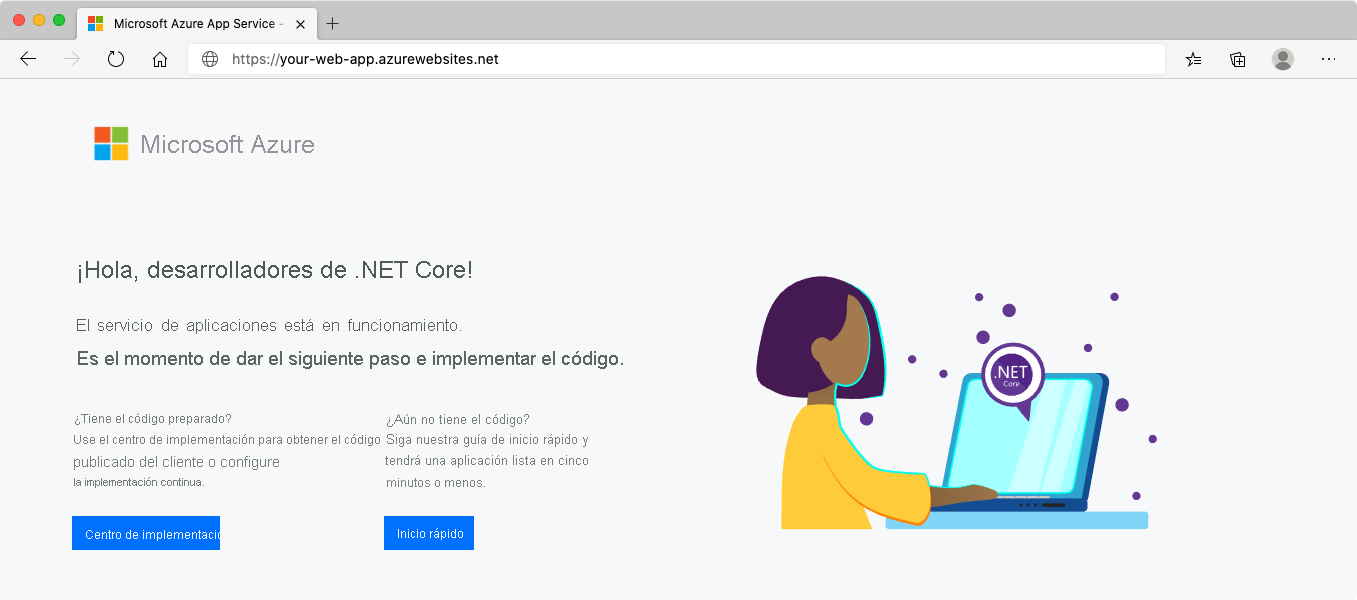
1. En el menú de Azure Portal o en la página de Inicio, seleccione Crear un recurso. Todo lo que se crea en Azure es un recurso. Aparece el panel de Marketplace.  
   Aquí, puede buscar el recurso que quiere crear o seleccionar uno de los recursos más populares que se crean en Azure Portal.
2. En el panel de menús de la izquierda, seleccione Web.
3. En el cuadro de búsqueda, busque y seleccione Aplicación web. Aparece el panel de recursos Crear aplicación web. Seleccione Crear.
4. En la pestaña Aspectos básicos, escriba los valores siguientes para cada opción.

| **Configuración** | **Valor** | **Detalles** |
| --- | --- | --- |
| Detalles del proyecto |  |  |
| Suscripción | Suscripción de Concierge | La aplicación web que va a crear debe pertenecer a un grupo de recursos. Aquí se selecciona la suscripción de Azure a la que pertenece el grupo de recursos (o a la que va a pertenecer, si se crea en el asistente). |
| Grupo de recursos | En el menú desplegable, seleccione [Grupo de recursos de espacio aislado]. | El grupo de recursos al que pertenecerá la aplicación web. Todos los recursos de Azure deben pertenecer a un grupo de recursos. |
| Detalles de instancia |  |  |
| Nombre | *Escriba un nombre único* | El nombre de la aplicación web. Este nombre formará parte de la dirección URL de la aplicación: *nombre\_de\_la\_aplicación*.azurewebsites.net. El nombre que elija debe ser único entre todas las aplicaciones web de Azure. |
| Publicar | Código | El método que se va a usar para publicar la aplicación. Al publicar la aplicación como código, también debe configurar Pila en tiempo de ejecución con el fin de preparar los recursos de App Service para ejecutar la aplicación. |
| Pila en tiempo de ejecución | .NET Core 3.1 (LTS) | Plataforma en la que se ejecuta la aplicación. La elección puede afectar a la disponibilidad de una opción de sistema operativo: en algunas pilas en tiempo de ejecución, App Service solo admite un sistema operativo. |
| Sistema operativo | Linux | El sistema operativo que se usa en los servidores virtuales que ejecutan la aplicación. |
| Region | Centro de EE. UU. | Región geográfica desde la que se va a hospedar la aplicación. |
| Plan de App Service |  |  |
| Plan de Linux | *Deje el valor predeterminado* | Nombre del plan de App Service que respaldará la aplicación. De forma predeterminada, el asistente creará un plan en la misma región que la aplicación web. |
| SKU y tamaño | F1 | El plan de tarifa del plan que se va a crear. Esto determina las características de rendimiento de los servidores virtuales en los que se basa la aplicación y las características a las que tiene acceso. Para seleccionar el nivel F1, seleccione Cambiar tamaño para abrir el asistente Selector de especificaciones. En la pestaña Desarrollo/pruebas, seleccione F1 en la lista y después Aplicar. |

1. 
2. Seleccione Revisar y crear para ir al panel de revisión y, para crear la aplicación, seleccione Crear.  
    Nota  
   La aplicación web puede tardar un minuto en crearse y estar lista para su uso.  
   En el portal se muestra el panel de implementación, donde se puede ver el estado de la implementación.

## **Vista previa de la aplicación web**

1. Una vez que la implementación se complete correctamente para la aplicación, seleccione Ir al recurso. El portal muestra el panel de información general de la aplicación web.  
   
2. Para obtener una vista previa del contenido predeterminado de la nueva aplicación web, seleccione su dirección URL en la parte superior derecha. La página de marcador de posición que se carga indica que la aplicación web está en funcionamiento y lista para recibir la implementación del código de la aplicación.



Deje abierta la pestaña del explorador con la página de marcador de posición de la nueva aplicación. Volverá a esa pestaña después de haber implementado la aplicación.

# **Preparación del código de la aplicación web**

Completado

100 XP

* 3 minutos

Elija el lenguaje de desarrollo

C#

Java

Node.js

Python

En esta unidad, aprenderá a crear el código para la aplicación web y a integrarlo en un repositorio de control de código fuente.

## **Arranque de una aplicación web**

Ahora que ha creado los recursos para implementar la aplicación web, tendrá que preparar el código que quiere implementar. Hay muchas maneras de arrancar una nueva aplicación web, por lo que es posible que lo que vea aquí sea diferente a lo que conoce. El objetivo es proporcionarle rápidamente un punto de partida para completar un ciclo completo hasta la implementación.

Nota

Todo el código y los comandos que se muestran en esta página solo se incluyen con fines explicativos; no es necesario ejecutar ninguno de ellos. Lo haremos en el ejercicio siguiente.

Puede usar la herramienta de línea de comandos del administrador de paquetes de Node (npm) que viene incluida con Node.js para crear una aplicación. El comando init crea un elemento package.json necesario para la aplicación:

Bash

Copiar

npm init -y

Si abre el archivo package.json que se ha creado en la carpeta actual, puede agregar una entrada start a la sección scripts para describir cómo se debe ejecutar la aplicación web:

JSON

Copiar

...

"scripts": {

"start": "node index.js"

},

...

Después, puede escribir el código real de la aplicación web en un nuevo archivo index.js:

JavaScript

Copiar

const http = require('http');

const port = process.env.PORT || 3000;

http.createServer(function(request, response) {

response.writeHead(200, { 'Content-Type': 'text/plain' });

response.end('Hello World!');

}).listen(port);

console.log(`Server running at http://localhost:${port}`);

Este código de ejemplo crea un servidor que escucha en el puerto 3000 del equipo y responde a todas las solicitudes con un mensaje "Hola mundo".

## **Adición del código al control de código fuente**

Una vez que el código de la aplicación web está listo, el paso siguiente suele consistir en incluirlo en un repositorio de control de código fuente, como Git. Si tiene Git instalado en el equipo, al ejecutar estos comandos en la carpeta de código fuente se inicializará el repositorio.

Bash

Copiar

git init

git add .

git commit -m "Initial commit"

Estos comandos le permitirán inicializar un repositorio de Git local y crear una primera confirmación con el código. Las ventajas inmediatas que obtiene es la capacidad de mantener un historial de los cambios con confirmaciones. Más adelante, también podrá sincronizar el repositorio local con un repositorio remoto, por ejemplo, hospedado en GitHub. Esto le permitirá configurar la integración continua y la implementación continua (CI/CD). Aunque se recomienda usar un repositorio de control de código fuente para las aplicaciones de producción, no es necesario para implementar una aplicación en Azure App Service.

Nota

El uso de CI/CD permite una implementación de código más frecuente de forma confiable, mediante la automatización de las compilaciones, las pruebas y las implementaciones para cada cambio de código. Permite ofrecer nuevas características y correcciones de errores para la aplicación de forma más rápida y eficaz.

# **Ejercicio: Escritura de código para implementar una aplicación web**

Completado

100 XP

* 5 minutos

Este módulo requiere un espacio aislado para completarse. Usó 1 de 10 espacios aislados por hoy. Mañana habrá disponibles más espacios aislados.

Activar espacio aislado

Elija el lenguaje de desarrollo

C#

Java

Node.js

Python

En esta unidad, se usarán herramientas de desarrollo para crear el código para una aplicación web de inicio.

## **Creación de un proyecto web**

Para crear una aplicación web Node.js de inicio, se usará el Administrador de paquetes de Node (npm) junto con código de JavaScript para ejecutar el procesamiento real de la página web.

1. Ejecute los comandos siguientes en Cloud Shell para crear un archivo package.json que describirá la aplicación Node.js.
2. Bash
3. Copiar

cd ~

mkdir helloworld

cd helloworld

npm init -y

Esto creará un archivo package.json en la carpeta actual. Debería verlo en la carpeta actual si escribe ls en la ventana del terminal. Necesitaremos un archivo de JavaScript para ejecutar la lógica del sitio web. Dado que este es solo un ejemplo básico, solo se necesitará un archivo index.js.

1. Ejecute el comando siguiente en el terminal para crear el archivo.
2. Bash
3. Copiar

touch index.js

1. Ahora habrá que realizar algunos cambios en los dos archivos. Escriba el comando siguiente en el terminal para abrir un editor interactivo.
2. Bash
3. Copiar

code .

1. Seleccione el archivo package.json y realice las modificaciones siguientes en la sección scripts para usar Node.js para iniciar la aplicación web.
2. JSON
3. Copiar

{

"name": "helloworld",

...

"scripts": {

"start": "node index.js"

},

...

}

1. Guarde el archivo mediante el menú … o la combinación de teclas (CTRL+S en Windows y Linux, Comando+S en macOS).

Importante

Cada vez que pegue o cambie código en un archivo en el editor, no olvide guardarlo posteriormente mediante el menú … o la tecla de aceleración (CTRL+S en Windows y Linux, o Comando+S en macOS).

1. Cambie al archivo index.js y agregue el siguiente contenido. Se trata de un pequeño programa de Node.js que siempre responde con "Hola mundo" cuando se realice cualquier solicitud GET al servidor.
2. JavaScript
3. Copiar

const http = require('http');

const server = http.createServer(function(request, response) {

response.writeHead(200, { "Content-Type": "text/html" });

response.end("<html><body><h1>Hello World!</h1></body></html>");

});

const port = process.env.PORT || 1337;

server.listen(port);

console.log(`Server running at http://localhost:${port}`);

1. Guarde el archivo y salga del editor. Para salir del editor, seleccione el menú … en la parte superior derecha o presione CTRL+Q.

### **Prueba opcional de la aplicación web**

Si abre una segunda sesión de shell de comandos, por ejemplo, si va hasta <https://shell.azure.com/>, puede probar la aplicación de forma local en Azure. Para ello, realice los pasos siguientes.

1. En la sesión de shell de comandos principal, ejecute los comandos siguientes para iniciar la aplicación web.
2. Bash
3. Copiar

cd ~/helloworld

npm start

1. En la segunda sesión de shell de comandos, ejecute el comando siguiente para ir a la aplicación web.
2. Bash
3. Copiar

curl http://127.0.0.1:1337/

1. Debe ver lo siguiente.
2. HTML
3. Copiar

<html><body><h1>Hello World!</h1></body></html>

1. En la sesión del shell de comandos principal, presione Ctrl+C para salir de la aplicación web.

# **Implementación de código en App Service**

Completado

100 XP

* 3 minutos

Ahora, veremos cómo se puede implementar la aplicación en App Service.

## **Implementación automatizada**

La implementación automatizada, o la integración continua, es un proceso que se usa para insertar nuevas características y correcciones de errores en un patrón repetitivo y rápido con un impacto mínimo en los usuarios finales.

Azure admite la implementación automatizada directamente desde varios orígenes. Están disponibles las opciones siguientes:

* Azure DevOps: Puede insertar el código en Azure DevOps (anteriormente conocido como Visual Studio Team Services), compilar el código en la nube, ejecutar las pruebas, generar una versión a partir del código y, por último, insertar el código en una aplicación web de Azure.
* GitHub: Azure admite la implementación automatizada directamente desde GitHub. Cuando conecte el repositorio de GitHub con Azure para la implementación automatizada, cualquier cambio que inserte en la rama de producción en GitHub se implementará de forma automática.
* Bitbucket: con sus similitudes con GitHub, puede configurar una implementación automatizada con Bitbucket.
* OneDrive: almacenamiento basado en la nube de Microsoft. Debe tener una cuenta de Microsoft vinculada a una de OneDrive para implementar en Azure.
* Dropbox: Azure admite la implementación desde Dropbox, que es un conocido sistema de almacenamiento basado en la nube parecido a OneDrive.

## **Implementación manual**

Hay algunas opciones que puede usar para insertar el código en Azure de forma manual:

* Git: App Service Web Apps incluyen una dirección URL de Git que se puede agregar como repositorio remoto. Al insertar en el repositorio remoto se implementará la aplicación.
* *az webapp up*: webapp up es una característica de la interfaz de línea de comandos az que empaqueta la aplicación y la implementa. A diferencia de otros métodos de implementación, az webapp up puede crear una aplicación web de App Service de forma automática si todavía no ha creado una.
* Implementación de archivo ZIP: Use az webapp deployment source config-zip para enviar un ZIP de los archivos de la aplicación a App Service. También se puede acceder a la implementación desde un archivo ZIP mediante utilidades básicas de HTTP, como curl.
* Implementación desde un archivo WAR: es un mecanismo de implementación de App Service diseñado específicamente para implementar aplicaciones web de Java mediante paquetes WAR. Se puede acceder a la implementación desde un archivo WAR mediante la API Kudu de HTTP, que se encuentra en http://<your-app-name>.scm.azurewebsites.net/api/wardeploy. Si este método no funciona, intente lo siguiente: https://<your-app-name>.scm.azurewebsites.net/api/wardeploy.
* Visual Studio: Visual Studio incluye un asistente para la implementación de App Service que le puede guiar a través del proceso de implementación.
* FTP/S: FTP o FTPS es una manera tradicional de insertar el código en muchos entornos de hospedaje, incluido App Service.

# **Ejercicio: Implementación del código en App Service**

Completado

100 XP

* 5 minutos

Este módulo requiere un espacio aislado para completarse. Usó 1 de 10 espacios aislados por hoy. Mañana habrá disponibles más espacios aislados.

Activar espacio aislado

Elija el lenguaje de desarrollo

C#

Java

Node.js

Python

En esta unidad, se implementará la aplicación web en App Service.

## **Implementación con az webapp up**

A continuación se implementará la aplicación Node.js con az webapp up. Este comando empaquetará la aplicación y la enviará a la instancia de App Service, donde se compilará e implementará.

En primer lugar, es necesario recopilar información sobre el recurso de aplicación web. Ejecute estos comandos para establecer las variables de shell que contienen el nombre de la aplicación, el nombre del grupo de recursos, el nombre del plan, la SKU y la ubicación. Usan otros comandos de az para solicitar la información de Azure; az webapp up necesita estos valores para dirigirse a la aplicación web existente.

Bash

Copiar

APPNAME=$(az webapp list --query [0].name --output tsv)

APPRG=$(az webapp list --query [0].resourceGroup --output tsv)

APPPLAN=$(az appservice plan list --query [0].name --output tsv)

APPSKU=$(az appservice plan list --query [0].sku.name --output tsv)

APPLOCATION=$(az appservice plan list --query [0].location --output tsv)

Ahora, ejecute az webapp up con los valores adecuados. Asegúrese de que está en el directorio helloworld antes de ejecutar este comando.

Bash

Copiar

az webapp up --name $APPNAME --resource-group $APPRG --plan $APPPLAN --sku $APPSKU --location "$APPLOCATION"

La implementación tardará un par de minutos, periodo durante el que verá el resultado del estado.

## **Comprobación de la implementación**

A continuación se desplazará a la aplicación para verla en activo. La última línea de salida de texto de az webapp up, antes de la salida JSON, tiene un vínculo a la aplicación. Haga clic en él para ir hasta allí en una nueva pestaña del explorador. La página tardará un momento en cargarse, ya que es la primera vez que App Service inicializa la aplicación.

A continuación, verá el mensaje de saludo de la aplicación, lo que indica que se ha implementado correctamente.