



## TUTORIAL

# Cómo usar la integración Git en Visual Studio Code

Git VS Code

By [James Quick](#)

Published on November 12, 2020 17.3k

Español

## Introducción

Visual Studio Code (VS Code) se ha convertido en uno de los editores más populares que existen para el desarrollo web. Ha conseguido tanta popularidad gracias a sus muchas funciones integradas, incluyendo la integración del Control de código fuente, es decir, con Git. Aprovechar el poder de Git desde VS Code puede hacer que su flujo de trabajo sea más eficiente y robusto.

En este tutorial, explorará usar la Integración del Control de código fuente con VS Code con Git.

## Requisitos previos

Para completar este tutorial, necesitará lo siguiente:

- Git instalado en su máquina. Para obtener más información sobre cómo hacer esto, revise el tutorial Primeros pasos con Git.
- La versión más reciente de Visual Studio Code instalada en su equipo.

## Paso 1: Familiarizarse con la pestaña Control de código fuente

Lo primero que debe hacer para aprovechar la integración del Control de código fuente es iniciar un proyecto como repositorio Git.

Abra Visual Studio Code y acceda al terminal integrado. Puede abrirlo usando el atajo de teclado `CTRL + `` en Linux, macOS o Windows.

En su terminal, cree un directorio para un nuevo proyecto y cambie a ese directorio:

```
$ mkdir git_test  
$ cd git_test
```

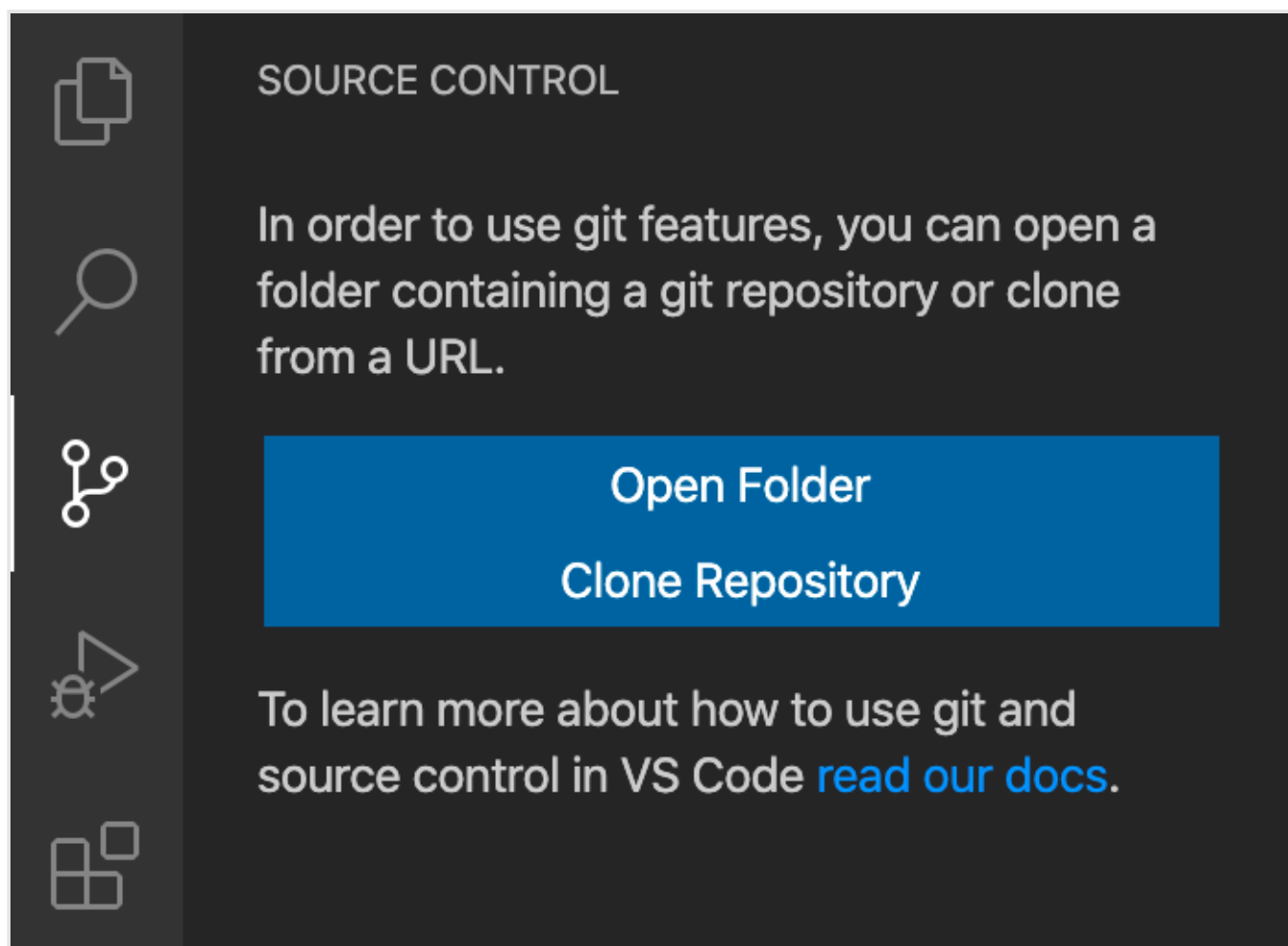
A continuación, cree un repositorio Git:

```
$ git init
```

Otra forma de conseguir esto con Visual Studio Code es abriendo la pestaña Control de código fuente (el icono parece una **brecha en la carretera**) en el panel izquierdo:

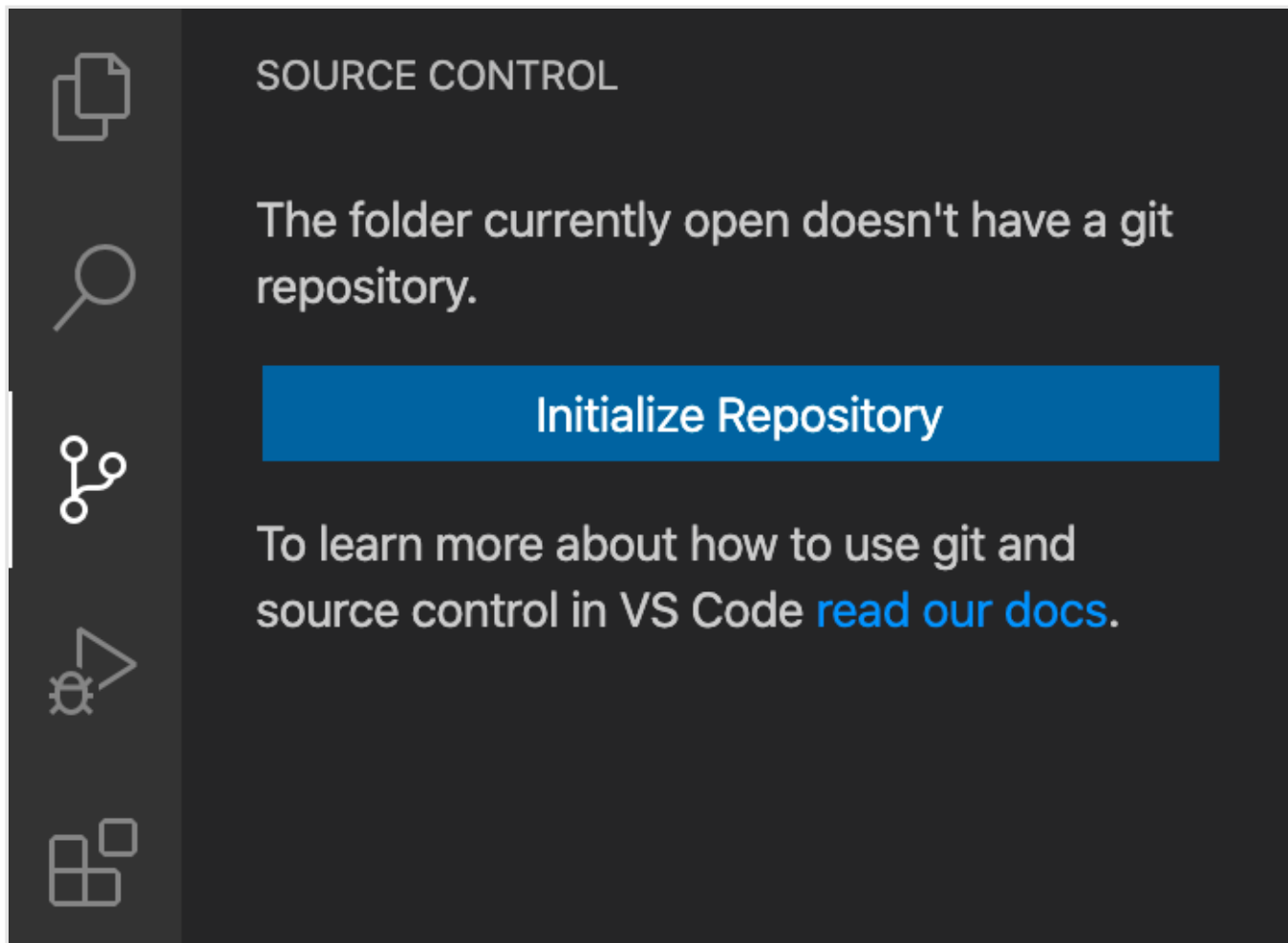


A continuación, seleccione **Abrir carpeta**:



Esto abrirá su explorador de archivos al directorio actual. Seleccione el directorio de proyecto preferido y haga clic en **Abrir**.

A continuación, seleccione **Iniciar repositorio**:



Si ahora comprueba su sistema de archivos, verá que incluye un directorio `.git`. Para hacer esto, use el terminal para navegar al directorio de su proyecto y liste todo su contenido:

```
$ ls -la
```

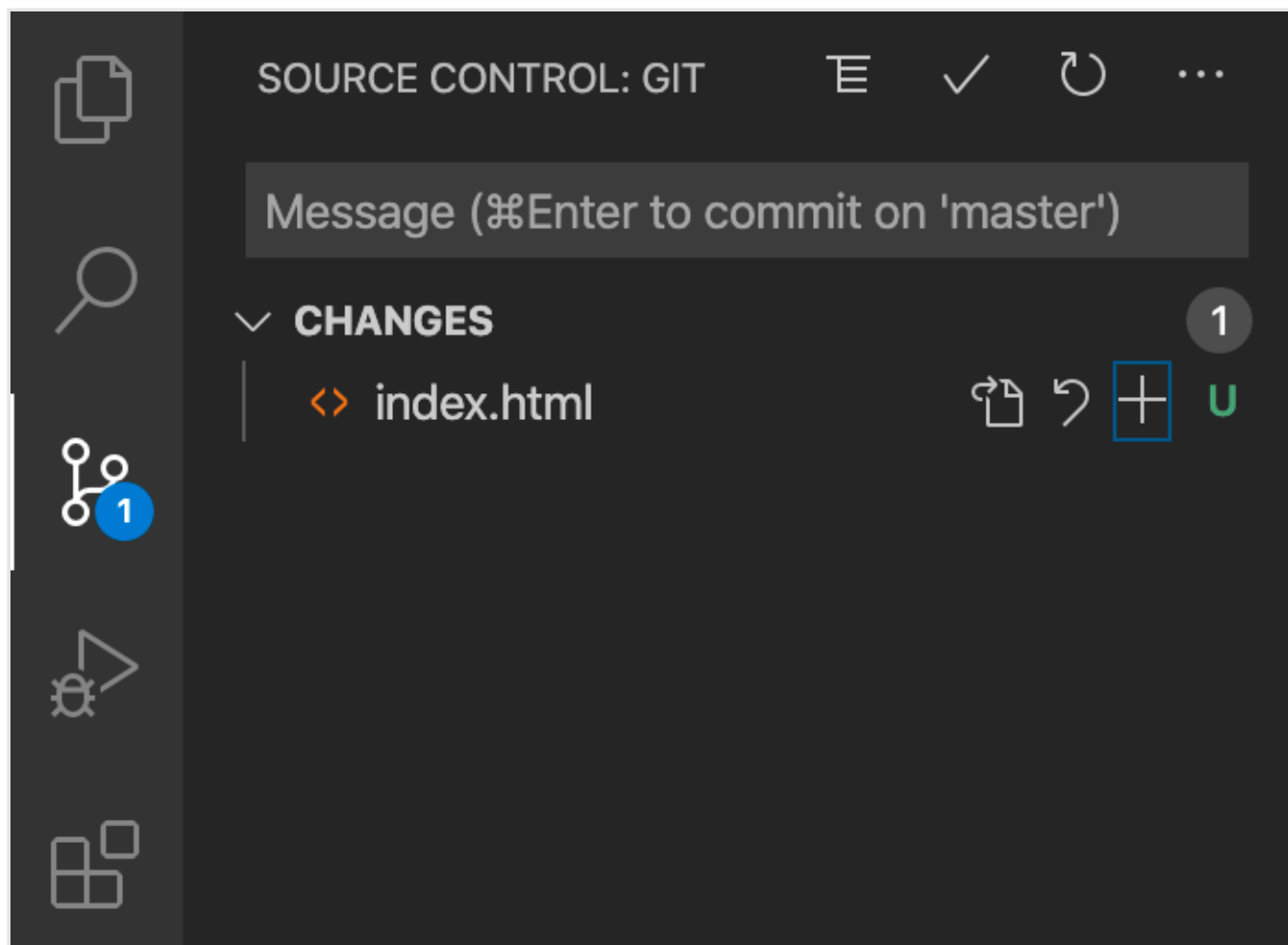
Verá el directorio `.git` que se creó:

Output

```
$ .  
$ ..  
$ .git
```

Ahora que se ha iniciado el repositorio, añada un archivo llamado `index.html`.

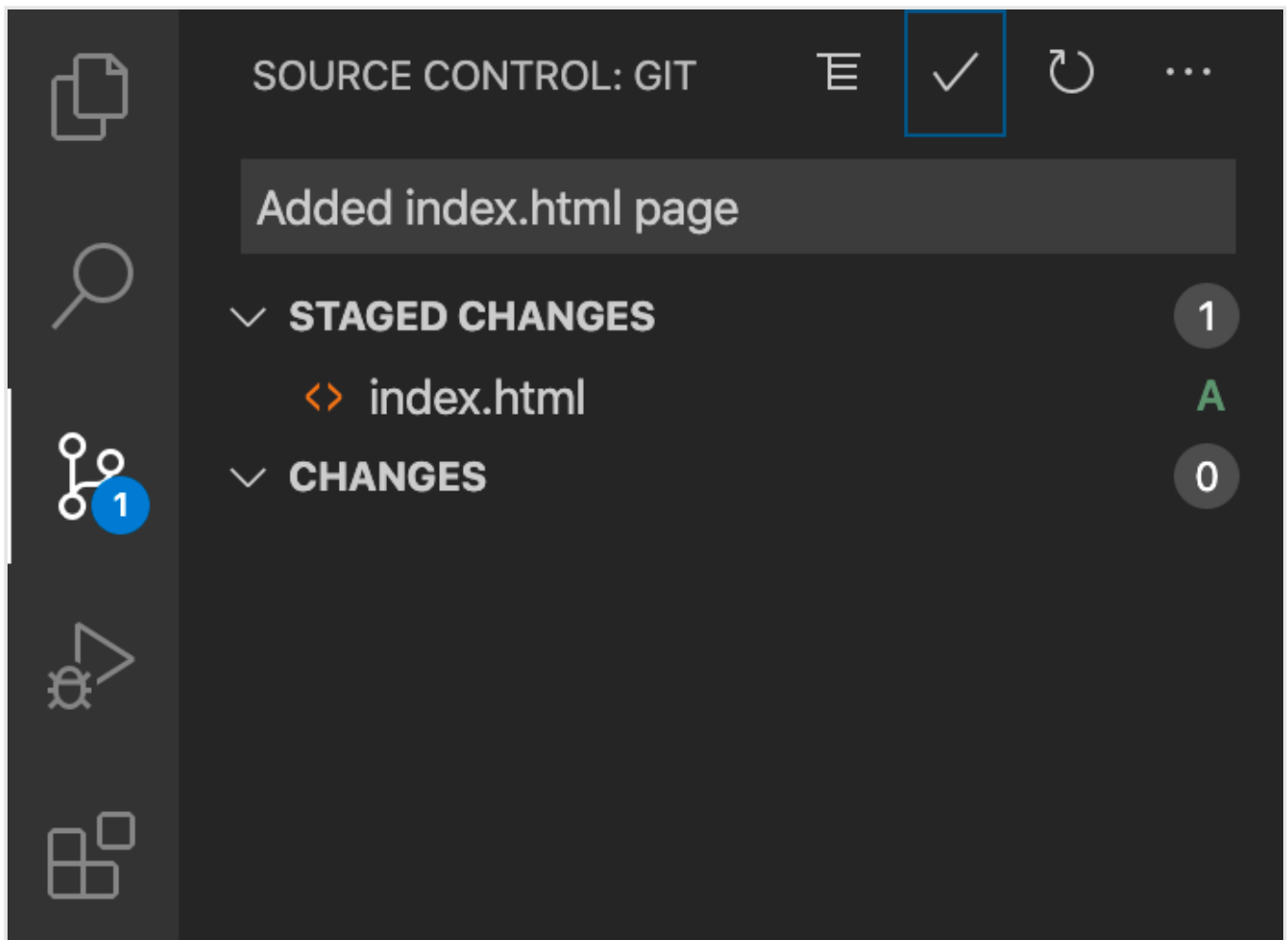
Tras hacer eso, verá en el **panel Control de código fuente** que su nuevo archivo se muestra con la letra **U** junto a él. **U** significa archivo sin seguimiento (*untracked file*), lo que significa que es nuevo o se ha cambiado, pero aún no ha sido añadido al repositorio:



Ahora puede hacer clic en el icono **plus (+)** junto al listado de archivos `index.html` para que el repositorio realice un seguimiento del archivo.

Una vez añadido, la letra junto al archivo cambiará a una **A**. **A** indica un nuevo archivo que se añadió al repositorio.

Para confirmar sus cambios, escriba un mensaje commit en el cuadro de entrada en la parte superior del **panel Control de código fuente**. A continuación, haga clic en el icono **check** para realizar la confirmación.

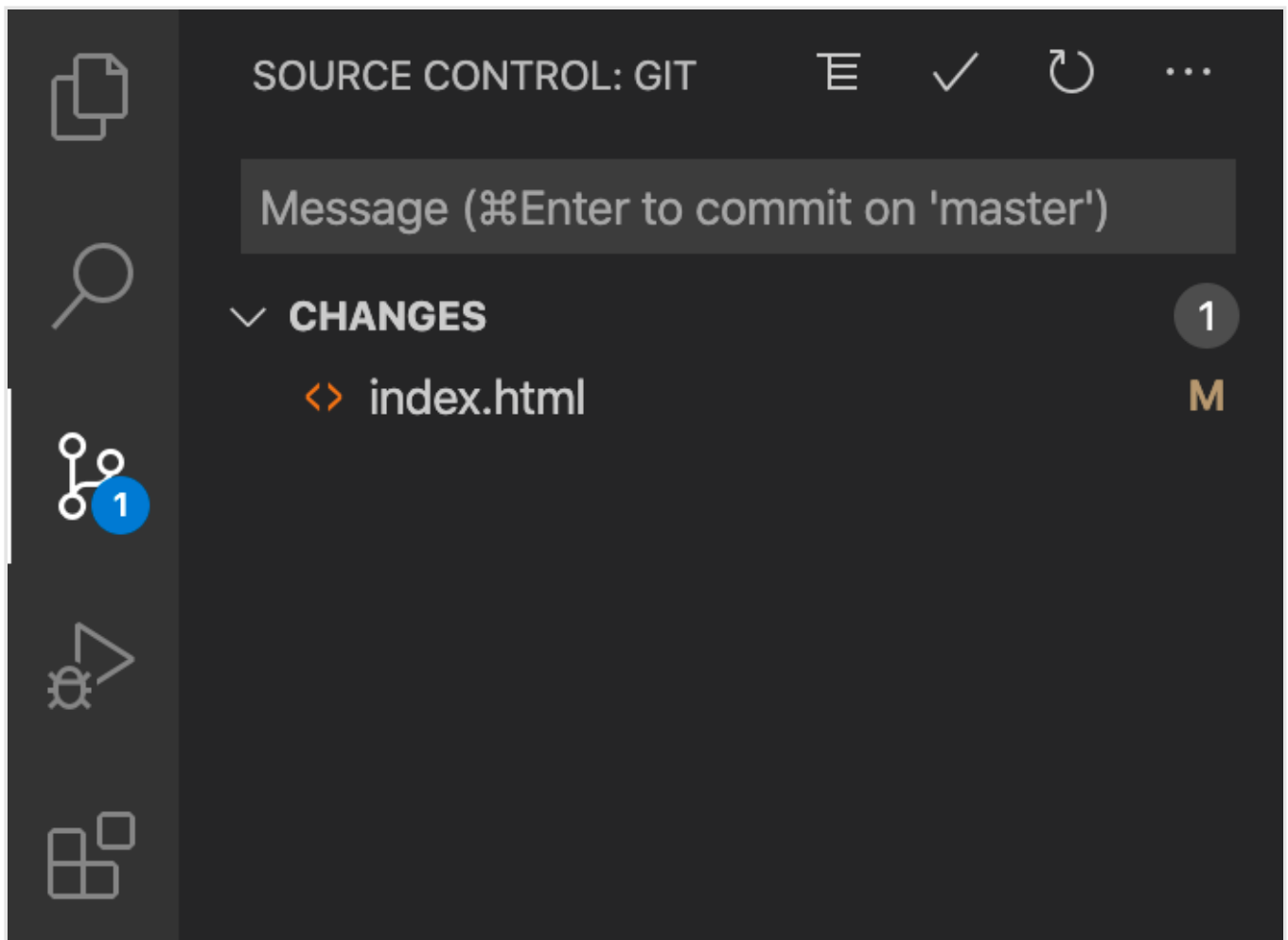


Tras hacerlo, observará que no hay cambios pendientes.

A continuación, añada un poco de contenido a su archivo `index.html`.

Puede usar un atajo Emmet para generar un esqueleto HTML5 en VS Code pulsando la tecla `!` seguida de la tecla `Tab`. Añada algo en el `<body>` como un encabezado `<h1>` y guárdelo.

En el panel de control de código fuente, verá que su archivo ha cambiado. Mostrará la letra **M** junto a él, que representa un archivo que se ha *modificado*.



Para practicar, confirme también este cambio.

Ahora que está familiarizado con la interacción con el panel de código fuente, pasará a interpretar los indicadores de margen.

## Paso 2: Interpretar indicadores de margen

En este paso, echará un vistazo a lo que se denomina el “*margen*” en VS Code. El margen es el área muy delgada a la derecha del número de línea.

Si ha usado code folding antes, los iconos **maximize** y **minimize** están en el margen.

Vamos a comenzar realizando un pequeño cambio en su archivo `index.html`, como un cambio en el contenido en la etiqueta `<h1>`. Tras hacerlo, observará una marca vertical azul en el margen de la línea que ha cambiado. La marca azul vertical significa que la línea correspondiente de código ha cambiado.

Ahora, intente eliminar una línea de código. Puede eliminar una de las líneas en la sección `<body>` de su archivo `index.html`. Observe que en el margen ahora hay un triángulo rojo. El triángulo rojo significa una línea o grupo de líneas que se han eliminado.

Finalmente, en la parte inferior de su sección `<body>` , añada una nueva línea de código y observe la barra verde. La barra verde vertical significa una línea de código que se ha añadido.

Este ejemplo muestra indicadores de margen para una línea modificada, una línea eliminada y una nueva línea:

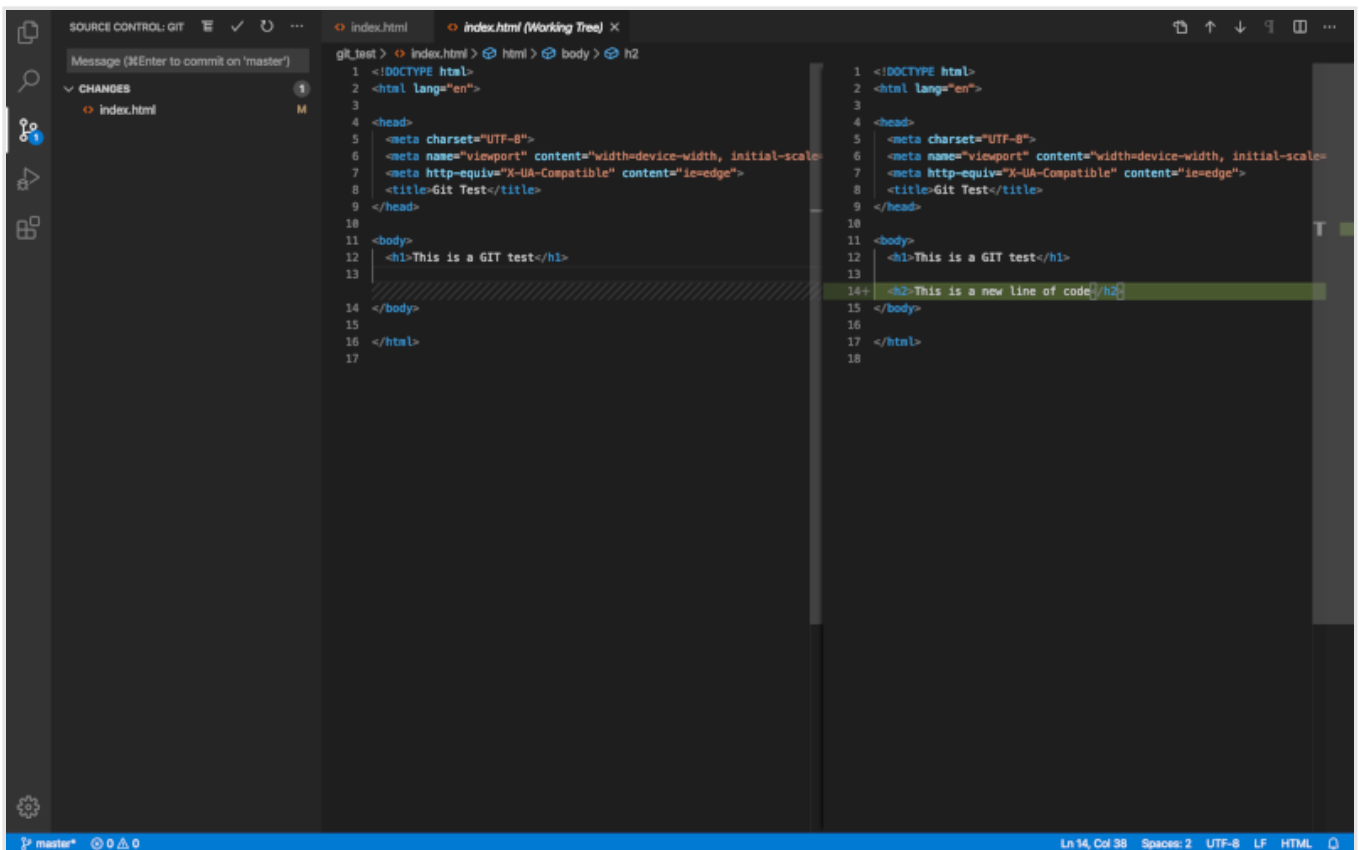
```
<> index.html X
git_test > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>Git Test</title>
9  </head>
10
11 <body>
12     <!-- the following line has been modified -->
13     <h1>This is an Updated GIT test</h1>
14
15     <!-- the following line has been removed -->
16
17     <!-- the following line has been added -->
18     <h2>This is a new line of code</h2>
19 </body>
20
21 </html>
```

### Paso 3: Diferenciar archivos

VS Code también tiene la capacidad de realizar un diff sobre un archivo. Normalmente, tendría que descargar una herramienta diff independiente para hacer esto, de forma que esta función integrada puede ayudarle a trabajar de forma más eficiente.

Para ver un diff, abra el panel de control del código fuente y haga doble clic en un archivo cambiado. En este caso, haga doble clic en el archivo `index.html` . Se lo dirigirá a una vista diff normal con la versión actual del archivo a la izquierda y la versión previamente confirmada del archivo a la derecha.

Este ejemplo muestra que se ha añadido una línea en la versión actual:

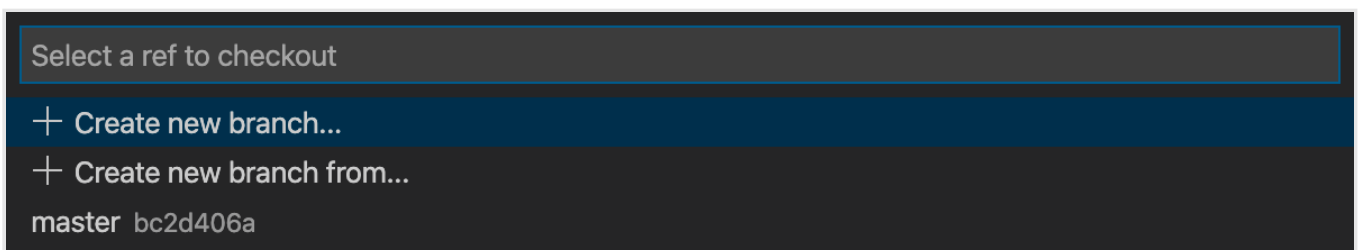


## Paso 4: Trabajar con ramas

Si se mueve a la barra inferior, tiene la capacidad de crear y cambiar ramas. Si echa un vistazo a la parte inferior izquierda del editor, debería ver el icono **control de código fuente** (el que parece una división en la carretera) seguido de probablemente el **master** o el nombre de la rama actualmente en funcionamiento.



Para crear una rama, haga clic en ese nombre de la rama. Debería aparecer un menú que le da la capacidad para crear una nueva rama:



Cree una nueva rama llamada `test`.

Ahora, realice un cambio a su archivo `index.html` que significa que está en la nueva rama `test`, como añadir el texto `this is the new test branch`.



Confirme esos cambios a la rama `test` . A continuación, haga clic en el nombre de la rama en la parte inferior izquierda de nuevo para volver a la rama `master` .

Tras volver a la rama `master` , observará que el texto `this is the new test branch` confirmado para la rama `test` ya no está presente.

## Paso 5: Trabajar con repositorios remotos

Este tutorial no tratará esto en profundidad, pero a través del panel Control de código fuente tiene acceso para trabajar con los repositorios remotos. Si ha trabajado con un repositorio remoto antes, observará comandos familiares como `pull`, `sync`, `publish`, `stash`, etc.

## Paso 6: Instalar extensiones útiles

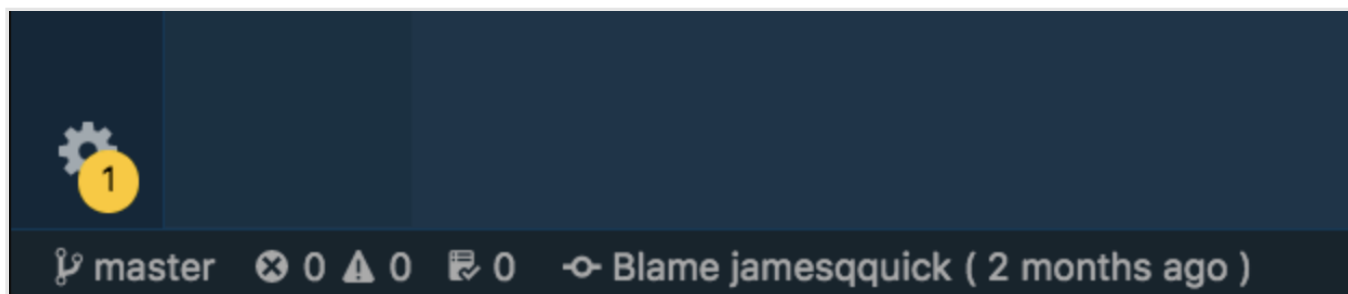
VS Code no solo cuenta con numerosas funciones integradas para Git, sino que también tiene algunas extensiones muy populares para añadir funciones adicionales.

### Git Blame

Esta extensión ofrece la capacidad de ver información Git Blame en la barra de estado para la línea seleccionada actualmente.

Esto puede sonar intimidante, pero no se preocupe, la extensión Git Blame es más sobre funcionalidad, no para hacer que alguien se sienta mal. La idea de “culpar” a alguien por un cambio de código no es para avergonzarlos, sino para averiguar quién es la persona adecuada a quien hacerle preguntas sobre ciertas piezas de código.

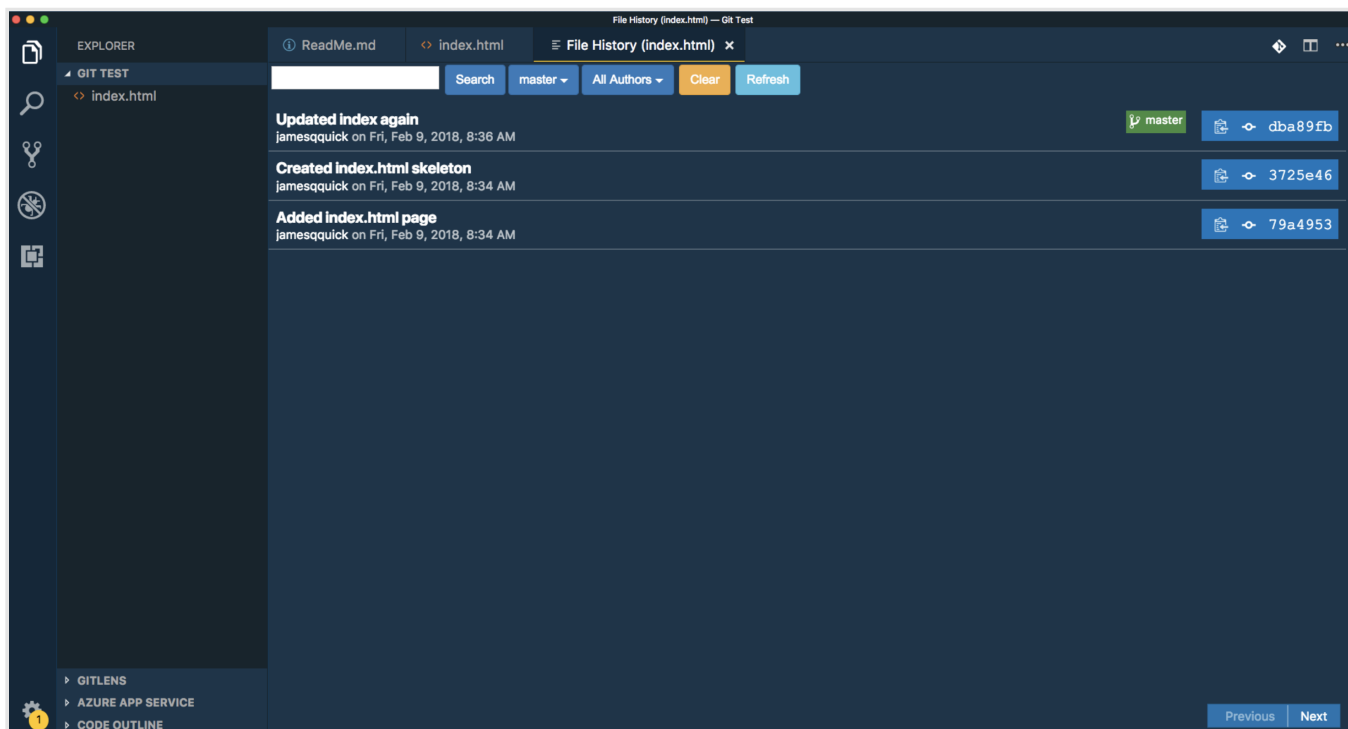
Como puede ver en la captura de pantalla, esta extensión proporciona un mensaje sutil relacionado con la línea de código actual sobre la que está trabajando en la barra de herramientas inferior, explicando quién realizó el cambio y cuándo lo realizó.



### Git History

Aunque puede ver los cambios actuales, realizar diffs, y administrar ramas con las funciones integradas en VS Code, no proporciona una vista profunda del historial de Git. La extensión Git History resuelve ese problema.

Como puede ver en la siguiente imagen, esta extensión le permite explorar de forma exhaustiva el historial de un archivo, un autor concreto, una rama, etc. Para activar la ventana Git History, haga clic con el botón derecho sobre un archivo y seleccione **Git: View File History**:



Además, puede comparar ramas y confirmaciones, crear ramas a partir de confirmaciones y mucho más.

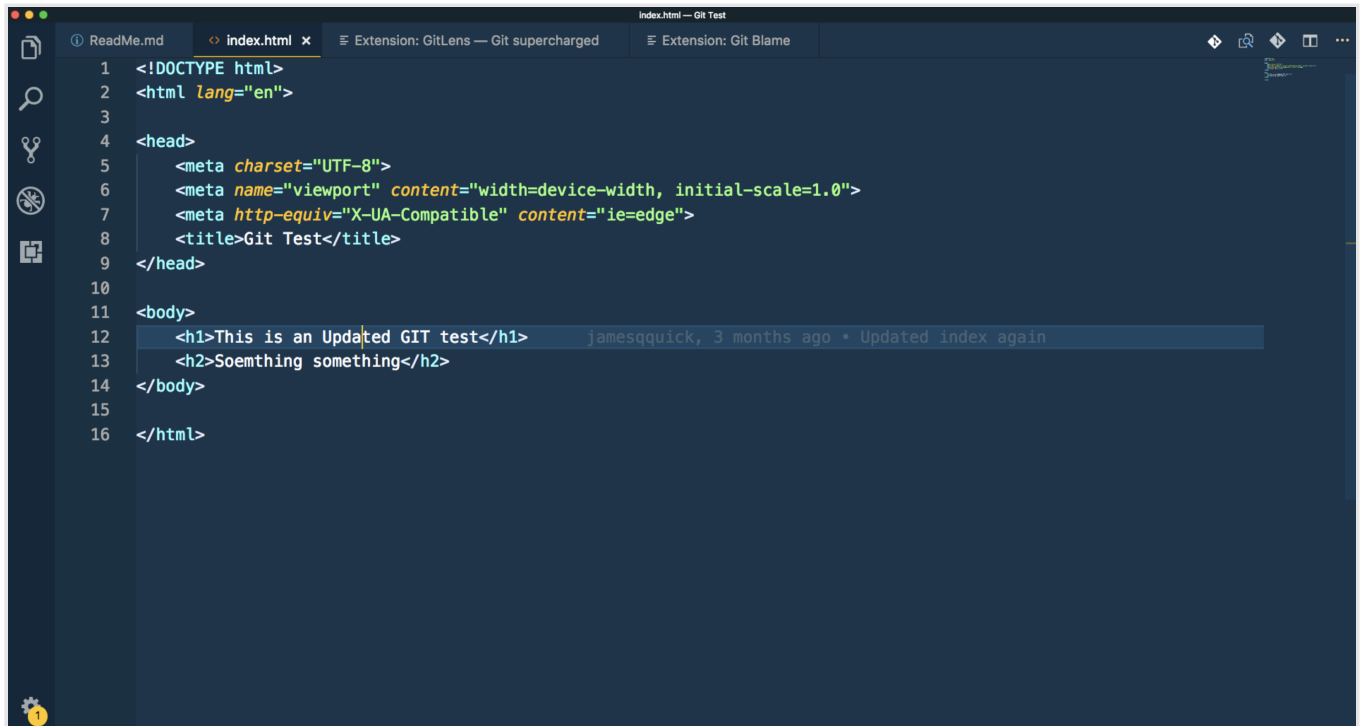
## Git Lens

GitLens sobrecarga las capacidades de Git integradas en Visual Studio Code. Lo ayuda a visualizar rápidamente la autoría del código mediante anotaciones de Git Blame y Code Lens, navegar de forma impecable y explorar repositorios de Git, obtener información valiosa mediante potentes comandos comparativos, y mucho más.

La extensión Git Lens es una de las más populares en la comunidad y también la más potente. En la mayoría de las maneras, puede sustituir a cada una de las dos extensiones anteriores con su funcionalidad.

Para la información “blame” aparece un mensaje sutil a la derecha de la línea en la que está trabajando actualmente para informarle de quién realizó el cambio, cuándo lo hizo y

el mensaje de confirmación asociado. Hay algunos fragmentos adicionales de información que aparecen cuando pasa el ratón sobre este mensaje como el cambio en el código, el sello de tiempo y mucho más.



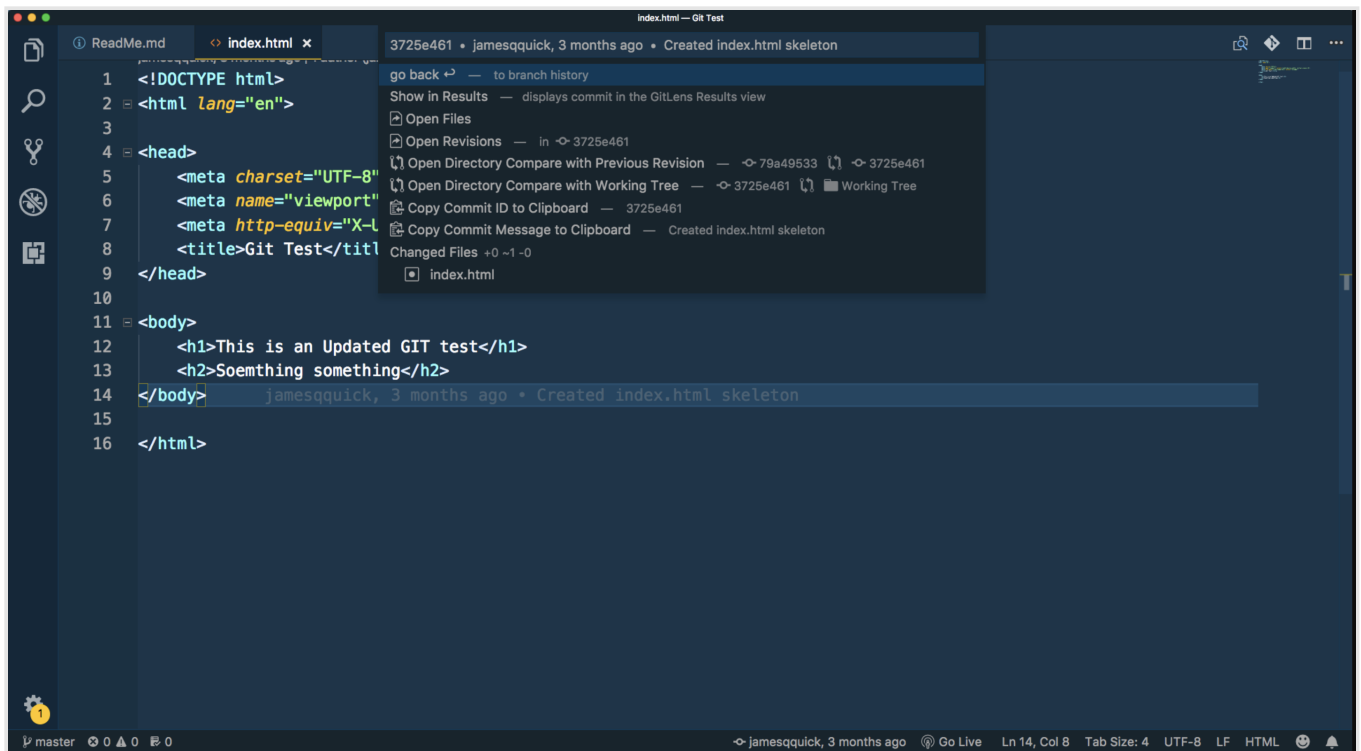
The screenshot shows the Visual Studio Code editor interface. The active file is 'index.html'. The code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta http-equiv="X-UA-Compatible" content="ie=edge">
8   <title>Git Test</title>
9 </head>
10
11 <body>
12   <h1>This is an Updated GIT test</h1>
13   <h2>Soemthing something</h2>
14 </body>
15
16 </html>
```

A tooltip is visible over line 12, displaying the commit message: 'jamesqquick, 3 months ago \* Updated index again'.

Para la información del historial de Git, esta extensión proporciona muchas funciones. Tiene acceso fácil a muchas opciones como mostrar el historial de archivos, realizar diffs con versiones anteriores, abrir una revisión específica y mucho más. Para abrir estas opciones, puede hacer clic en el texto en la barra de estado inferior que contiene el autor que editó la línea de código y hace cuánto se editó.



Esto abrirá la siguiente ventana:



Esta extensión cuenta con numerosas funciones y tardará un tiempo en aprender todo lo que ofrece.

### Conclusión





En este tutorial, exploré cómo usar la integración de control de código fuente con VS Code. VS Code puede gestionar muchas funciones que previamente habrían requerido la descarga de una herramienta independiente.

¿Qué calidad tuvo la traducción?  

Was this helpful? 

Yes

No

[Report an issue](#)

### About the authors



James Quick



**James Quick**

Developer. Speaker. Teacher.  
Developer Advocate at Auth0.

## Still looking for an answer?



Ask a question



Search for more help

### RELATED

#### Join the DigitalOcean Community



##### Join 1M+ other developers and:

- Get help and share knowledge in Q&A
- Subscribe to topics of interest
- Get courses & tools that help you grow as a developer or small business owner

Join Now

Cómo formatear código con Prettier en Visual Studio Code

[Tutorial](#)

Cómo instalar Git en Ubuntu 20.04

[Tutorial](#)

### Comments

## 0 Comments

Leave a comment...

Sign In to Comment



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.



HOLLIE'S HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth?  
We'd like to help.



#### BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

Featured on [Community](#) [Kubernetes Course](#) [Learn Python 3](#) [Machine Learning in Python](#)  
[Getting started with Go](#) [Intro to Kubernetes](#)

DigitalOcean Products [Virtual Machines](#) [Managed Databases](#) [Managed Kubernetes](#) [Block Storage](#)  
[Object Storage](#) [Marketplace](#) [VPC](#) [Load Balancers](#)

## Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

[Learn More](#)





SPACES (3)

jon-db

https://jon-db.nyc3.digitaloceanspaces.com

...



© 2021 DigitalOcean, LLC. All rights reserved.

## Company

[About](#)

[Leadership](#)

[Blog](#)

[Careers](#)

[Partners](#)

[Referral Program](#)

[Press](#)

[Legal](#)

[Security & Trust Center](#)

## Products

[Pricing](#)

[Products Overview](#)

[Droplets](#)

[Kubernetes](#)

[Managed Databases](#)

[Spaces](#)

[Marketplace](#)

[Load Balancers](#)

[Block Storage](#)

[API Documentation](#)

[Documentation](#)

[Release Notes](#)

## Community

[Tutorials](#)

[Q&A](#)

[Tools and Integrations](#)

[Tags](#)

[Product Ideas](#)

[Write for DigitalOcean](#)

[Presentation Grants](#)

[Hatch Startup Program](#)

[Shop Swag](#)

[Research Program](#)

[Open Source](#)

[Code of Conduct](#)

## Contact

[Get Support](#)

[Trouble Signing In?](#)

[Sales](#)

[Report Abuse](#)

[System Status](#)