**Tutorial DevOps con Azure**

Contents

[Introduccion 1](#_Toc23174858)

[Descarga de Herramientas 1](#_Toc23174859)

[Crear cuenta en Microsoft Azure 4](#_Toc23174860)

[Activar el Powershell 5](#_Toc23174861)

[Creación del proyecto en Azure 6](#_Toc23174862)

[Subir el proyecto a Azure 7](#_Toc23174863)

[Azure PIPELines 9](#_Toc23174864)

# Introduccion

En este documento se hará una breve ejemplificación de cómo llevar una aplicación de un framework, en este caso Laravel, a ser desarrollada de forma continua. De forma general se harán los siguientes pasos: Descargar las herramientas necesarias, crear la cuenta en Microsoft Azure, creación del proyecto en GitHub, activar el Powershell, creación del proyecto en Azure, subir el proyecto a Azure, y por último crear el pipeline para así lograr la integración continua.

# Descarga de Herramientas

Para esta demostración se hará el ejemplo con Laravel, es un framework para desarrollar aplicaciones y servicios web con PHP. Para lograr esto debemos realizar las siguientes descargas:

* PHP (7.2) - <https://www.php.net/downloads.php>
* Composer - <https://getcomposer.org/download/>
* FileZilla - <https://filezilla-project.org/download.php?platform=win64>
* Wamp (Opcional) - <https://sourceforge.net/projects/wampserver/>

Principalmente es necesario descargar PHP para que la aplicación y sus dependencias puedan ser descargadas y ejecutadas. FileZilla nos permite conectarnos a servidores por medio de SSH y manipular de forma gráfica los diferentes ficheros que estén alojados en el servidor. Compecer es un manejador de dependencias de PHP el cual nos permitirá crear el proyecto base de Laravel. Por otro lado, es opcional la descarga de Wamp debido a que solo se usa para poder visualizar la aplicación de forma local.

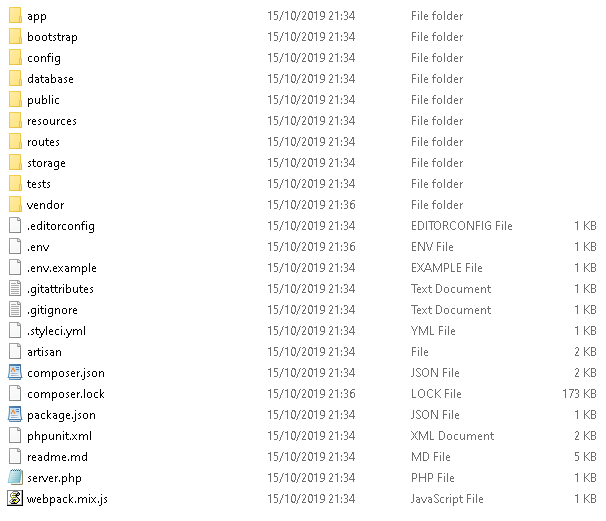
Instalación de Laravel, correr en la consola personal:

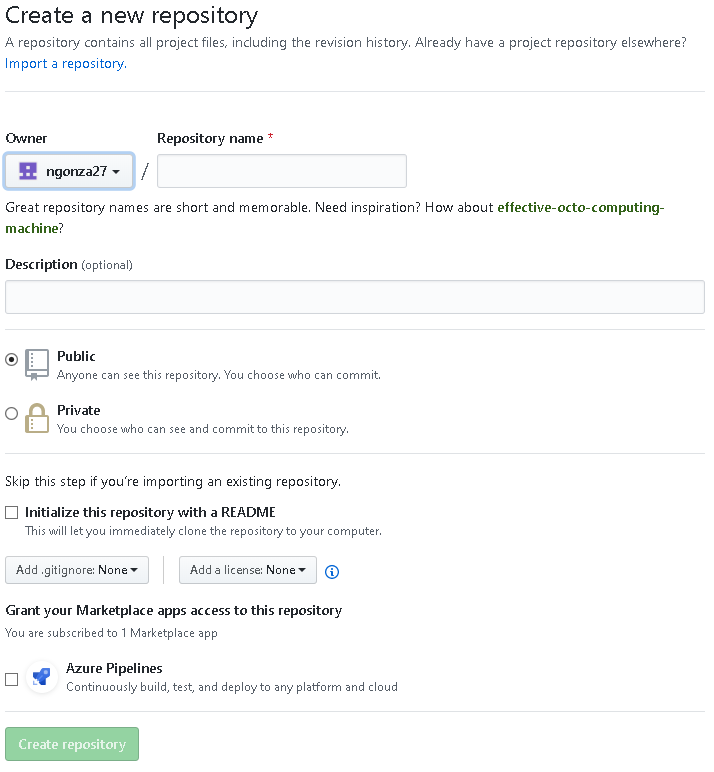
$ composer global require laravel/installer

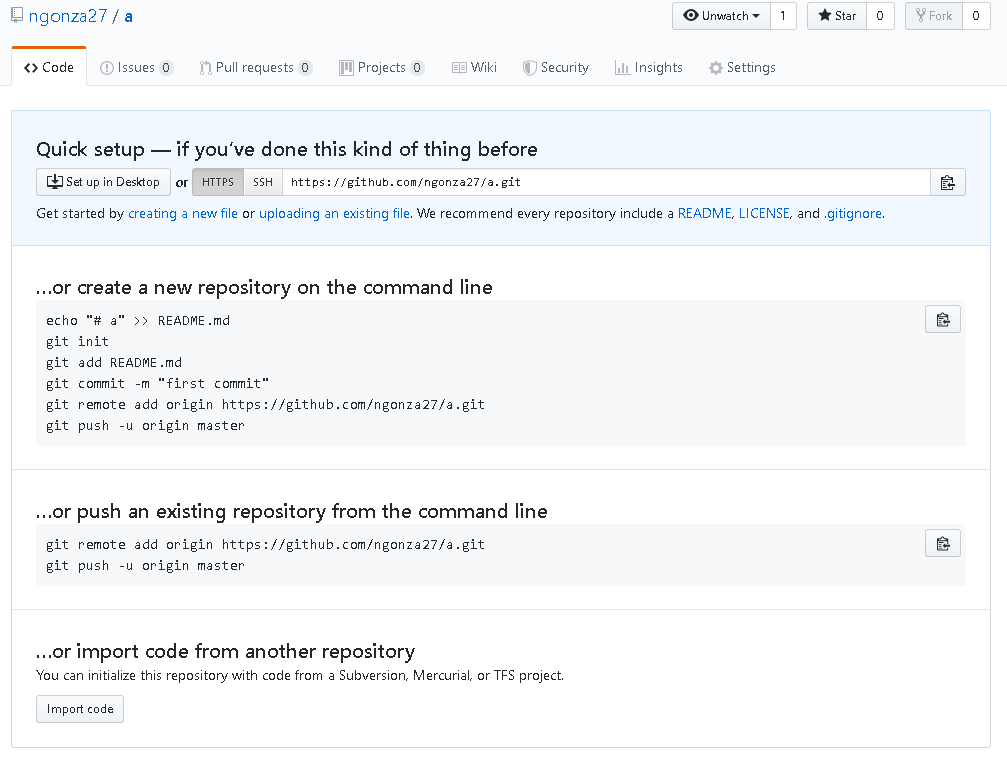
Una vez descargado e instalado los programas, situese en su carpeta de preferencia y ejecute este comando en la consola de su computador para crear el proyecto:

$ composer créate-project laravel/Laravel <nombre\_proyecto>

De haber ejecutado el comando correctamente, debería aparecer la siguiente jerarquía de ficheros:



Ingrese a su cuenta de GitHub. De no tener una registre una cuenta en el siguiente enlace: <https://github.com/join?source=header-home>. Cree un nuevo repositorio con el nombre el cual le asigno al proyecto de Laravel.



En la consola de **su** computador personal, acceda a la carpeta donde genero su proyecto de Laravel y realice los siguientes comandos:

Inicialice el proyecto

$ git init

Agregue el código del proyecto

$ git add .

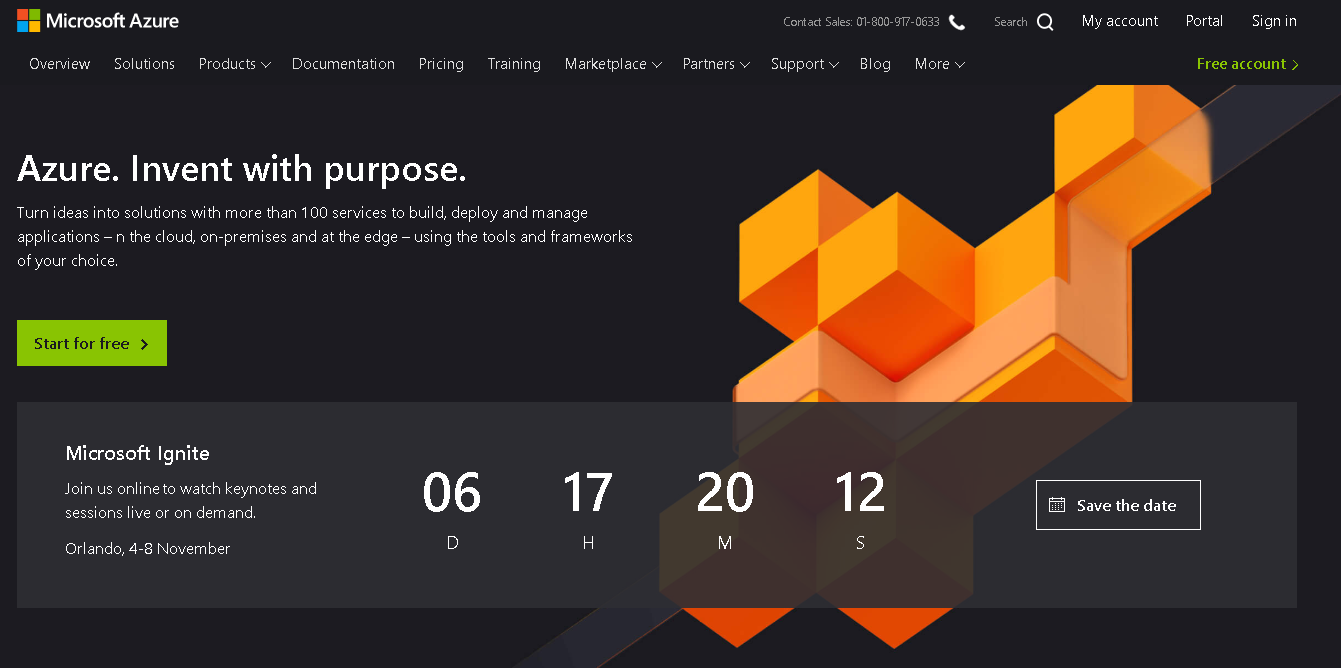
Confirme los cambios

$ git commit -m “<comentario>”

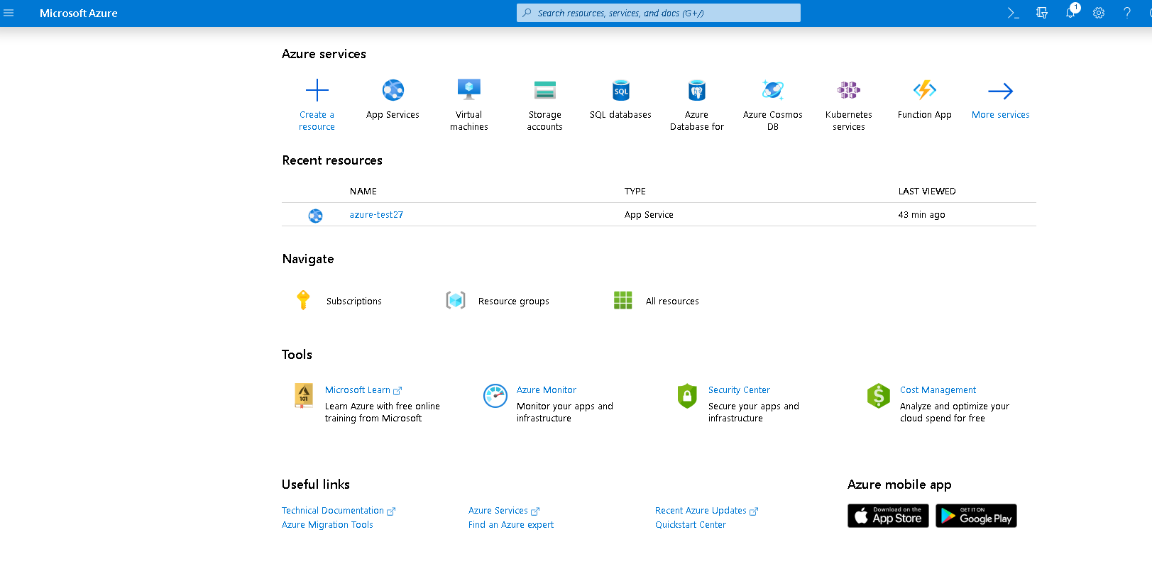
Agregue su código a GitHub

$ git push -u origin master

# Crear cuenta en Microsoft Azure

Abrir su navegador de preferencia e ingrese al siguiente enlace: <https://azure.microsoft.com/en-gb/>. Al estar en la siguiente interfaz, ingrese a la opción de **Start for free**, lo llevara a otra pagina y ingrese de nuevo a la opción **Start for free.**

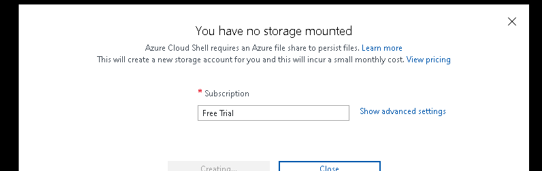
Regístrese con el correo institucional de la universidad. Una vez creada la cuenta ingrese al Portal principal.

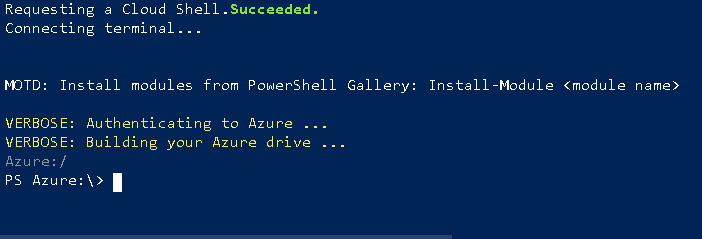


# Activar el Powershell

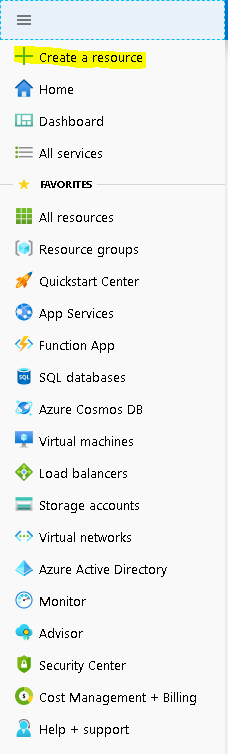
Acceda a la terminal. Selección la opción Powershell y cree una Storage account por defecto.



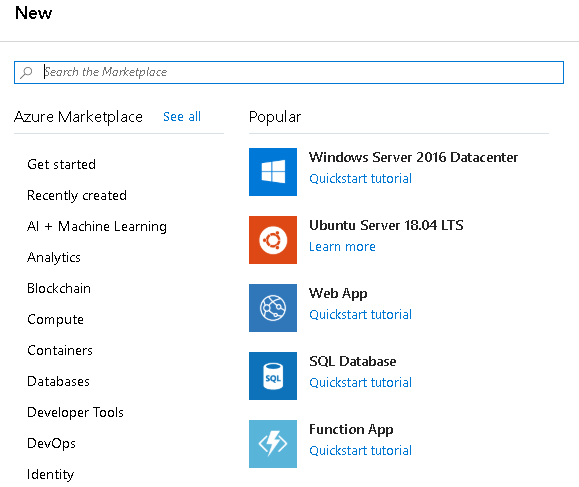


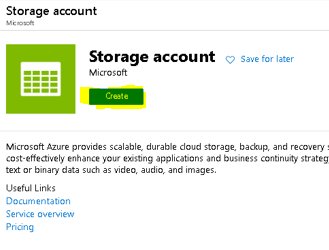
Una vez creado el storage account y terminado el proceso de la creación de la terminal, al acceder a ella nuevamente, debería observar algo parecido:

De no haberle funcionado, acceda al menú en la parte superior izquierda y seleccione Create a resource.



En la barra de búsqueda, ingrese Storage Account y cree un nuevo volumen de almacenamiento con las opciones por defecto que ofrece Microsoft Azure.





# Creación del proyecto en Azure

Al tener disponible la terminal, ingrese a ella y ejecute los siguientes comandos:

**Importante** las secciones resaltadas en amarillo dependen de cada proyecto, ¡los usuarios, contraseñas y nombres pueden variar por cada proyecto!

Inicializar el servidor:

$ az webapp deployment user set --user-name <usuario> --password <contraseña>

Localizar físicamente el servidor, para el tutorial se usó “West Europe”, pero escoger entre las diferentes ubicaciones: <https://azure.microsoft.com/es-es/global-infrastructure/regions/>

$ az group create --name <myResourceGroup> --location "West Europe"

Creación de la aplicación-web, de ser el caso cambiar “PHP” por el lenguaje del framework que usen para su proyecto:

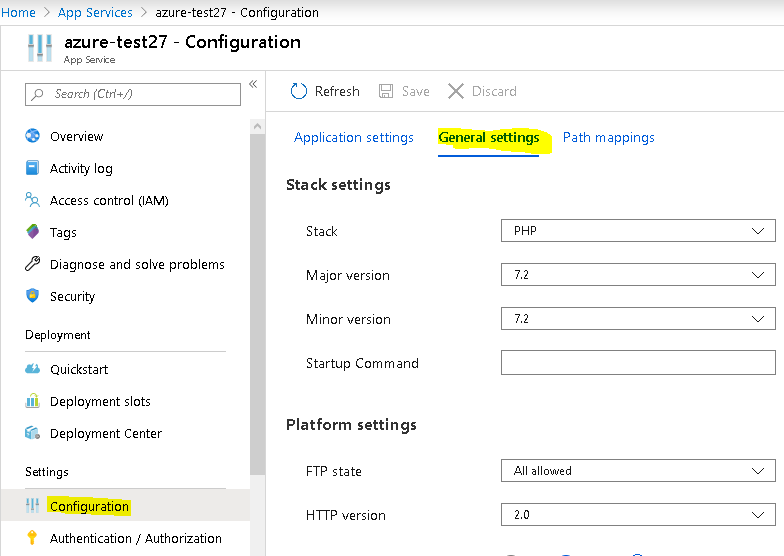
$ az appservice plan create --name <myAppServicePlan> --resource-group <myResourceGroup> --sku B1 --is-linux

$ webapp create --resource-group myResourceGroup --plan myAppServicePlan --name <nombreProyecto> --runtime "PHP|7.0" --deployment-local-git

En la consola se le mostrará este mensaje: Local git is configured with url of 'https://ngonza27@azure-test27.scm.azurewebsites.net/azure-test27.git' Note que para su proyecto el URL será diferente. Esto muestra donde su aplicación será alojada, en este caso: <http://azure-test27.azurewebsites.net>.

**Importante**

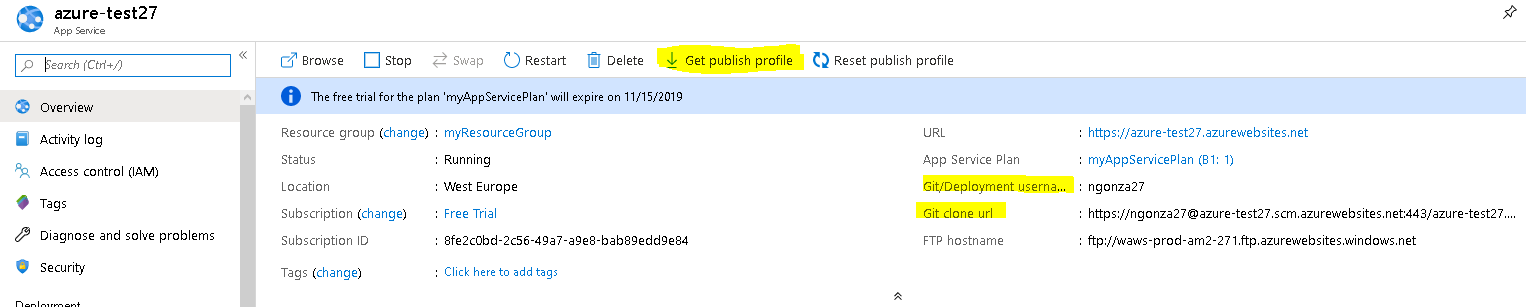
Ingrese a App Services y seleccione la aplicación que usted creó. Y verifique que el lenguaje que usted selecciono es el correcto al igual que la versión la cual este trabajando, para este tutorial se esta usando PHP 7.2.



# Subir el proyecto a Azure

Descargue el publish profile (Get publish profile) y guárdelo en su computador en un lugar el cual pueda acceder fácilmente. En la consola de **su** computador personal, acceda a la carpeta donde genero su proyecto de Laravel y realice los siguientes comandos:

**Importante** Al ingresar a **App Services** de **su aplicación** seleccione **Overview**. Ahí podrá observar información importante, ahí podrá encontrar el **Git clone url** el cual se usa para agregar los proyectos a un repositorio alojado en el servidor creado previamente.



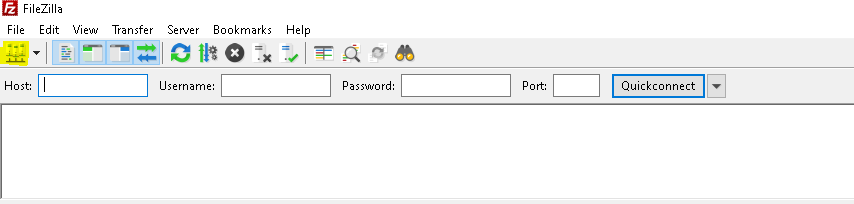
Agrega su Proyecto al repositorio alojado en Microsoft Azure

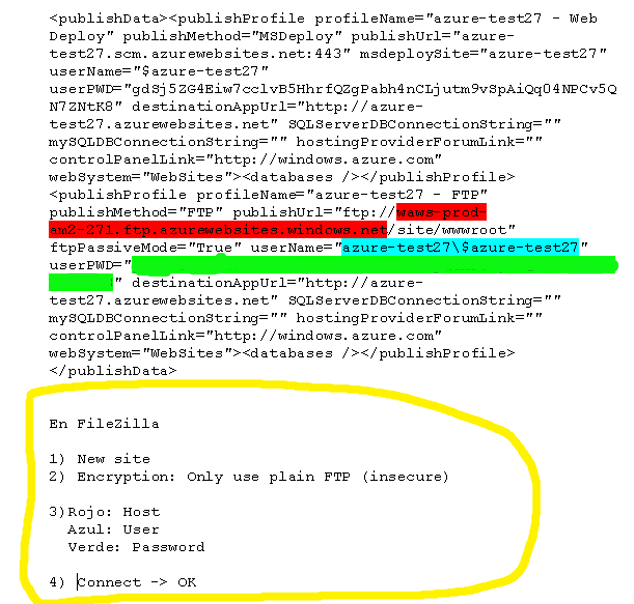
$ git remote add azure <Git clone url>

Agregue su código a Azure

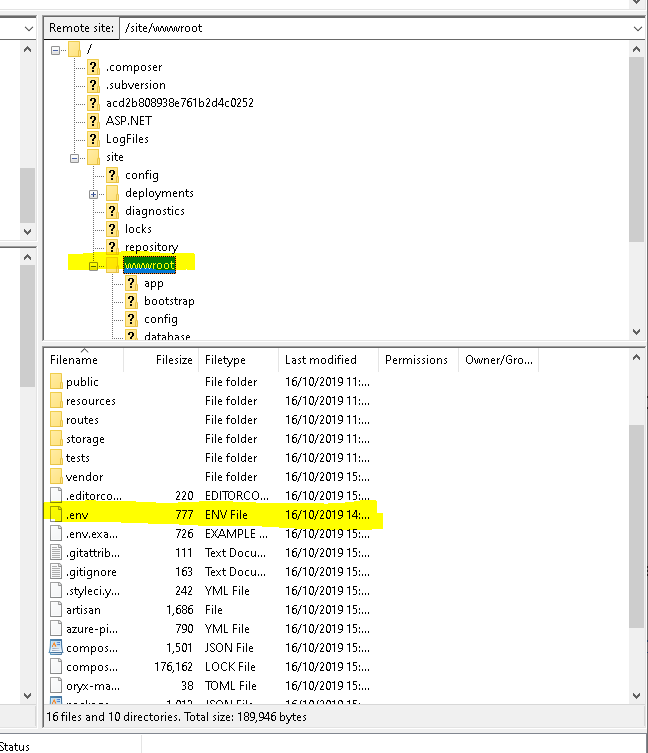
$ git push azure master

**Opcional (Solo para el tutorial de Laravel)**

Abra FileZilla y seleccione el siguiente icono:

Abrir el Archivo descargado(<nombre>.PublishSettings) y llenar los campos del numeral 3 de la siguiente foto:

Al ingresar a su proyecto por medio de FileZilla, ingrese a la carpeta principal del proyecto **wwwroot:**



Agregue a **wwwroot** un nuevo fichero llamado **.env** el cual contiene lo siguiente:

APP\_NAME=Laravel

APP\_ENV=local

APP\_KEY=

APP\_DEBUG=true

APP\_URL=http://localhost

LOG\_CHANNEL=stack

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=laravel

DB\_USERNAME=root

DB\_PASSWORD=

BROADCAST\_DRIVER=log

CACHE\_DRIVER=file

QUEUE\_CONNECTION=sync

SESSION\_DRIVER=file

SESSION\_LIFETIME=120

REDIS\_HOST=127.0.0.1

REDIS\_PASSWORD=null

REDIS\_PORT=6379

MAIL\_DRIVER=smtp

MAIL\_HOST=smtp.mailtrap.io

MAIL\_PORT=2525

MAIL\_USERNAME=null

MAIL\_PASSWORD=null

MAIL\_ENCRYPTION=null

AWS\_ACCESS\_KEY\_ID=

AWS\_SECRET\_ACCESS\_KEY=

AWS\_DEFAULT\_REGION=us-east-1

AWS\_BUCKET=

PUSHER\_APP\_ID=

PUSHER\_APP\_KEY=

PUSHER\_APP\_SECRET=

PUSHER\_APP\_CLUSTER=mt1

MIX\_PUSHER\_APP\_KEY="${PUSHER\_APP\_KEY}"

MIX\_PUSHER\_APP\_CLUSTER="${PUSHER\_APP\_CLUSTER}"

# Azure PIPELines

En el siguiente link se encuentra de forma resumida como asociar un repositorio de GitHub con los Pipelines de Azure:

<https://geeks.ms/jorge/2019/02/26/como-asociar-un-proyecto-open-source-de-github-en-azure-devops/>