# **Como Implementar un contenedor en App Service via Vs Code**

## Introducción:

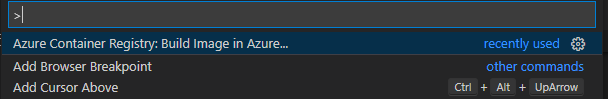
En el presente documento, vamos a realizar la publicación, a través de Visual Code, de una Docker Image sobre una App Service.

Se mostrará los pasos desde la creación del Azure Container Registry hasta el consumo de la App Service con la obtención de los datos.

En esta ocasión, todas las creaciones necesarias se realizarán en el Visual Studio Code

* Requisitos:
* Api creada sobre Flask o Django
* Extensión de Azure y Docker Instaladas sobre VS Code.
* DockerFile creado en la App
* Ambiente (Env) creado con los paquetes necesarios y generar el archivo yml.
* Repositorio en Azure DevOps.
* Creación del Azure Container Registry y construcción de la Imagen de Docker

Dentro del Visual Code, hacer Ctrl + Shif + P, en la ventana emergente seleccionar:

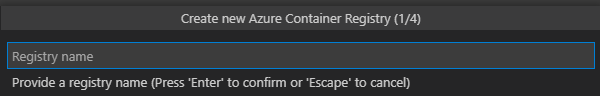


Posteriormente se elige en RunId, para la Docker Image (por lo general se usa el automático), luego la suscripción del recurso.

Lo siguiente que se mostrara es una ventana asi:

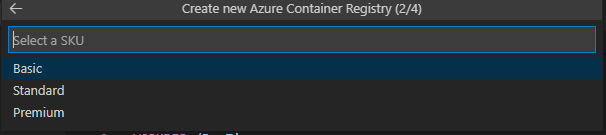


En esta ventana podrá seleccionar crear un nuevo ACR, o bien construir una nueva imagen sobre un acr creado previamente. En nuestro caso vamos a crear uno:



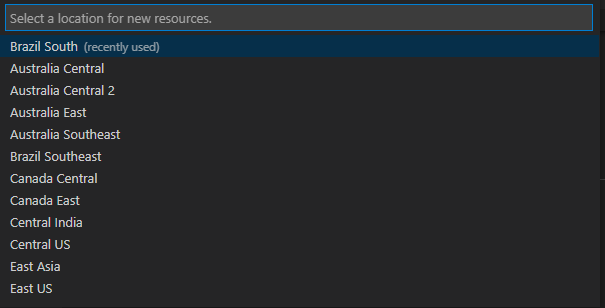
Para el nombre, se recomienda usar uno intuitivo, por ejemplo iniciar con las acr(nombre elegido para nuestro acr), en algunas ocasiones es aconsejable diferenciar entre otras cosas, la suscripción, si es para test o producción, etc. *Nota: solo se puede utilizar caracteres alfanuméricos.*

El siguiente paso será elegir el precio para nuestro recurso, cabe destacar que la variación en los diferentes precios se da por almacenamiento, webhooks y replicación grafica. Para nuestro ejemplo, se elige Basic por conveniencia

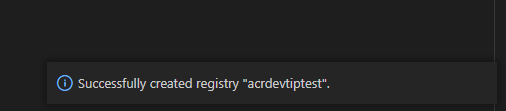
**

*Nota: para más información sobre los tipos de precios https://azure.microsoft.com/es-es/pricing/details/container-registry/*

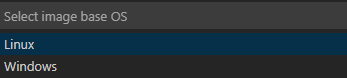
Lo siguiente será elegir el grupo de recursos, ya sea crear uno nuevo, o bien utilizar alguno previamente creado. Finalmente se elige la ubicación del recurso, en nuestro caso utilizamos Brazil South, pero queda a criterio del usuario, teniendo en cuenta ubicación del grupo de recursos, disponibilidad, etc.

**

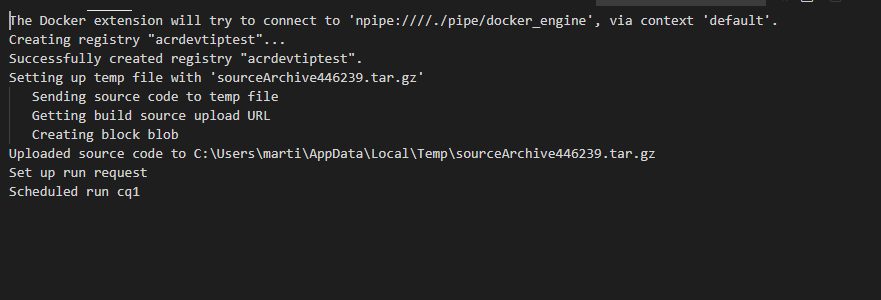
¡¡¡YA TENEMOS NUESTRO ACR CREADO!!!



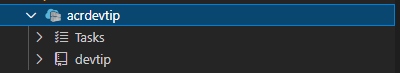
Ahora, debemos construir la imagen de Docker (Docker Image), se debe elegir que SO, tendrá la imagen, en nuestro caso elegimos Linux, pero queda a criterio del usuario



Se comenzará a crear la imagen correspondiente, este proceso puede tardar algunos minutos



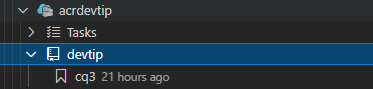
Para validar la correcta creación de la imagen, hay que ir a la extensión de Docker > Registries > Azure > Mi suscripción > Acr recientemente creado

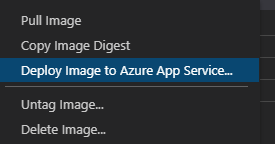


* Creación de App Service y publicación de Docker Image

Primer paso, ir a la extensión de Docker > Registries > Azure > Mi suscripción > Acr recientemente creado > Docker Image > Buscar la última imagen creada y hacer botón

derecho sobre la misma

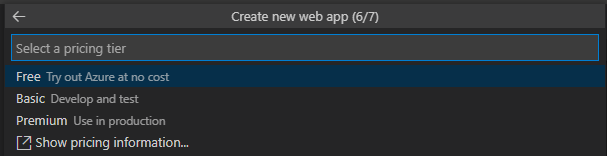




Aquí se deberá elegir suscripción, y nombre de la AppService, se recomienda iniciar con api-nombre elegido, grupo de recursos (se puede crear uno).

Posteriormente lo que se elige es el App Service Plan, también al igual que los grupos de recursos se puede crear uno o usar un plan existente.

Ahora toca elegir el precio, nosotros para esta prueba elegimos el Free por conveniencia, pero el precio va a depender del criterio, ya sea por espacio de disco, máximo de instancias, auto escalamiento, entre otros



*Nota para mayor detalle de los precios:* [*https://azure.microsoft.com/en-us/pricing/details/app-service/windows/*](https://azure.microsoft.com/en-us/pricing/details/app-service/windows/)

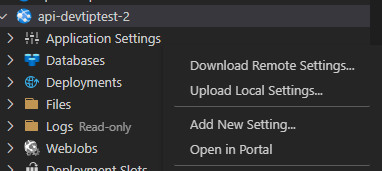
Finalmente se elige la ubicación del recurso, en nuestro caso utilizamos Brazil South, pero queda a criterio del usuario, teniendo en cuenta ubicación del grupo de recursos, disponibilidad, etc. Inicia la creación de la AppService

Una vez finalizada correctamente la creación se obtendrá el URL, para el consumo de los datos

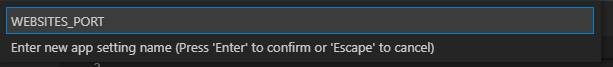


* Configuración de los puertos (IMPORTANTE)

Este es el último paso para configuración de la app service. Nos ubicamos en la extensión de Azure > App Service > nuestra app > Application Settings. Hacemos botón derecho y elegimos “Add New Settings”

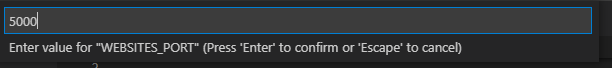


Se mostrara un menú como el siguiente :



Se deberá agregar una configuración llamada ***WEBSITES\_PORT*** (sin espacios al final)

El ultimo paso es configurar el puerto correspondiente, esto varía dependiendo si la api esta sobre Flask (puerto 5000) o Django (8000).



Una vez realizado esto, se recomienda reiniciar el AppService desde el portal de azure (a veces no es necesario).

Posteriormente, probar consumir el AppService (via postman por ejemplo)

**FIN!!**