**Ejercitación Azure CAMP**

Guía de ejercicios

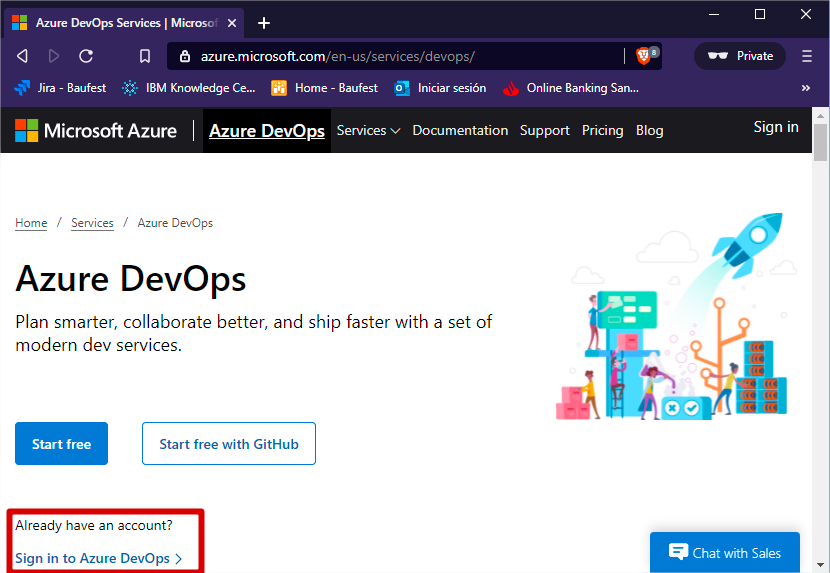
**|Baufest**

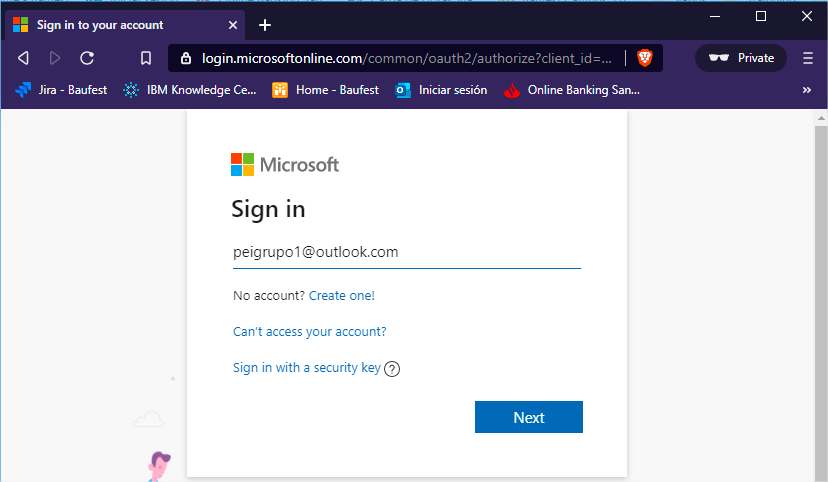
mayo de 2021

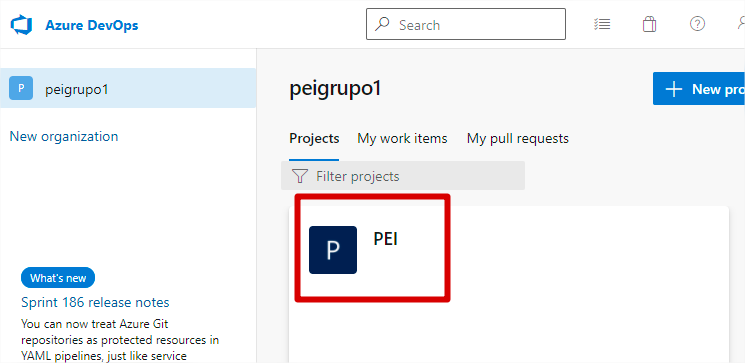
**Guía Práctica Despliegue en de container en WebApps**

1. **Ingresar a Azure Devops:**

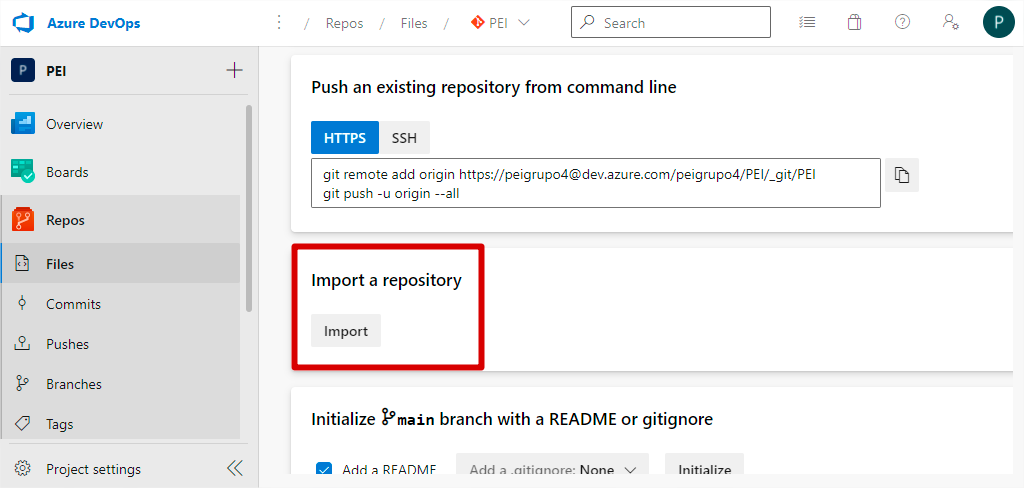
* URL: https://dev.azure.com
* Mail: [peigrupo1@baufest.com](mailto:peigrupo1@baufest.com)



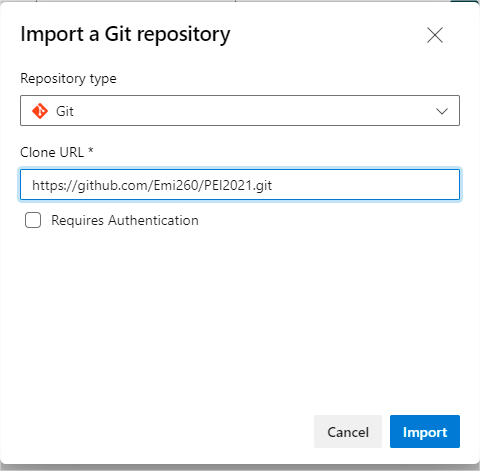




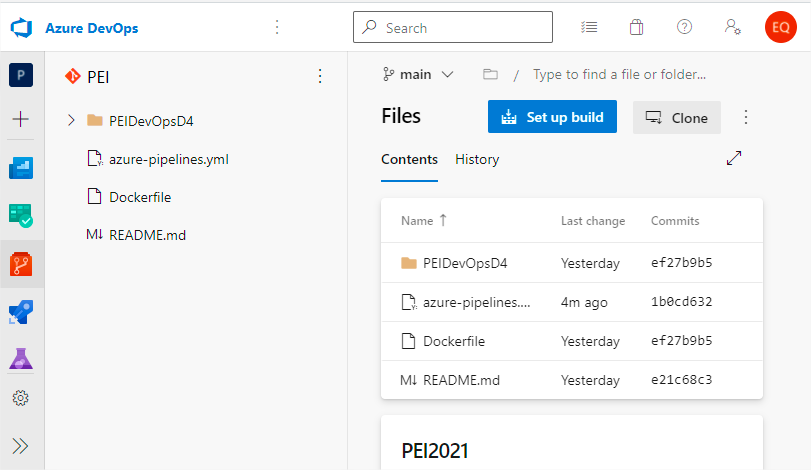
1. **Importamos el repositorio de GitHub al proyecto:**
   1. Nos dirigimos a la opción de repositorios y vamos a la opción de import



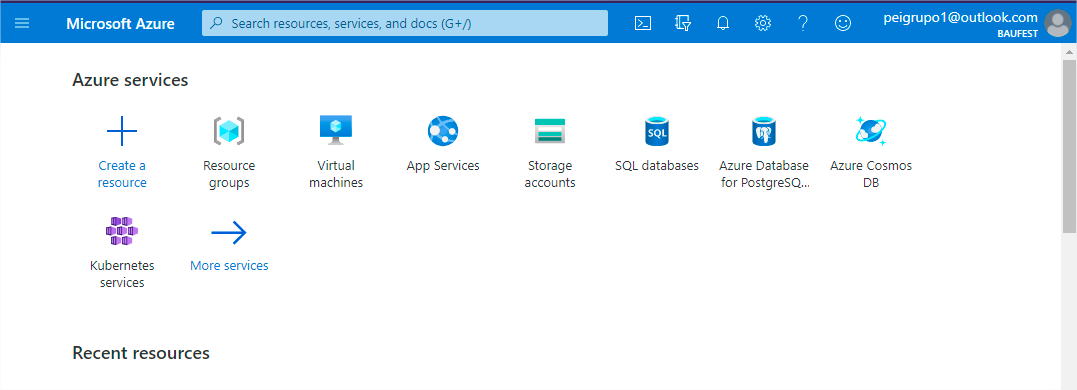
* 1. Copiamos la URL del repo de GitGub “https://github.com/Emi260/PEI2021.git” y hacemos clic en “Import”



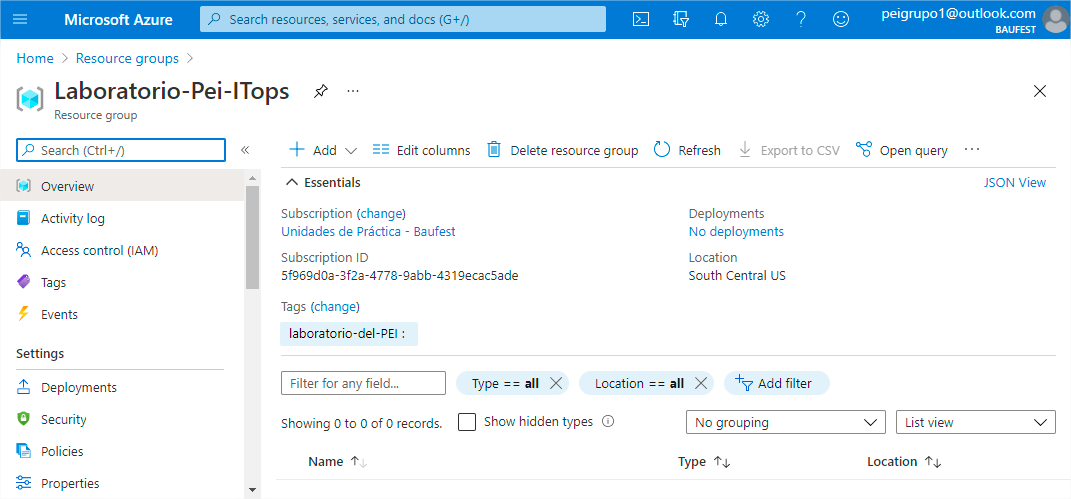
* 1. Si todo salió bien vamos a ver el repositorio con los archivos importados:



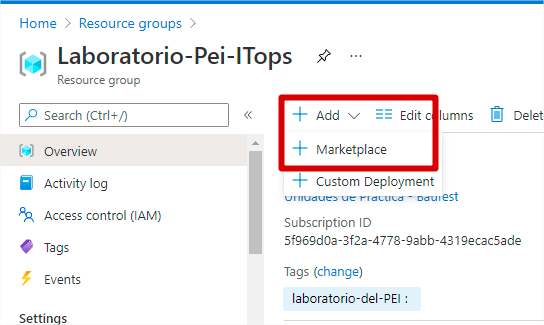
1. **Ingresar al portal de Azure para crear los recursos que vamos a utilizar:**
   1. Ingresamos al portal de Azure <https://portal.azure.com/>, con el mismo usuario y clave del paso 1.



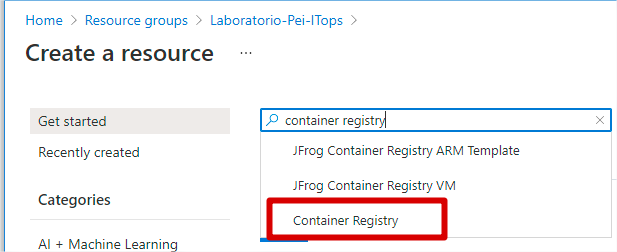
* 1. Ingresamos a “Resource groups” e ingresamos al recurso “[Laboratorio-Pei-ITops](https://portal.azure.com/#@baufestcloud.onmicrosoft.com/resource/subscriptions/5f969d0a-3f2a-4778-9abb-4319ecac5ade/resourceGroups/Laboratorio-Pei-ITops)”



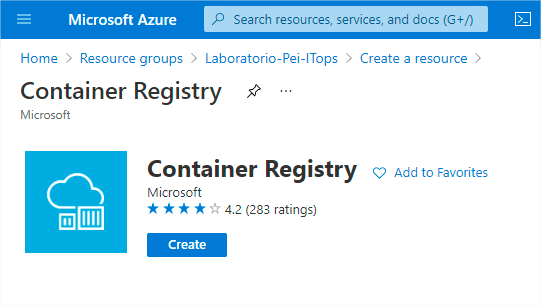
* 1. Abrimos el Marketplace



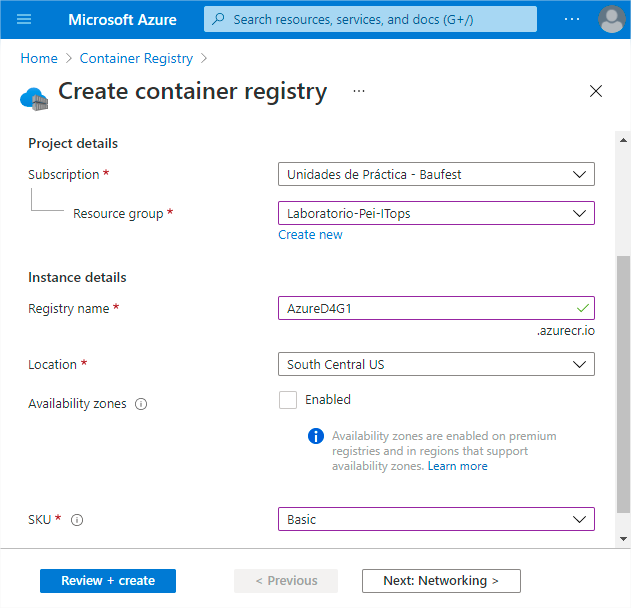
1. Creación del Container Registry:
   1. Ponemos en la barra de busqueda “container registry”:



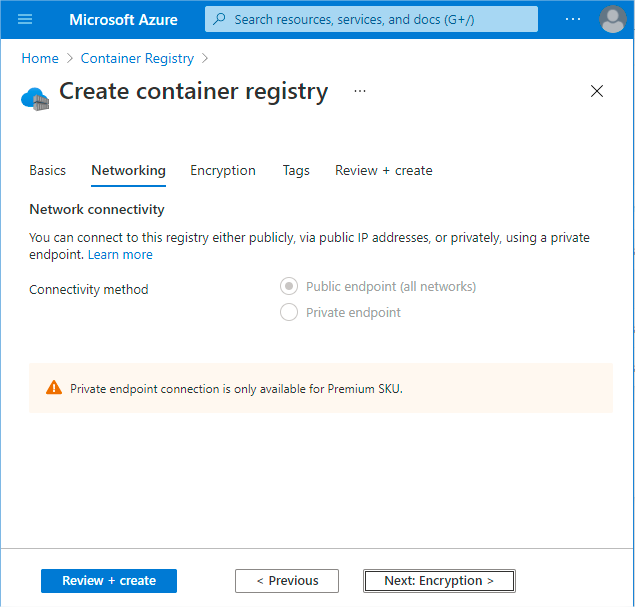
* 1. En la siguiente pantalla hacemos clic en “Create”



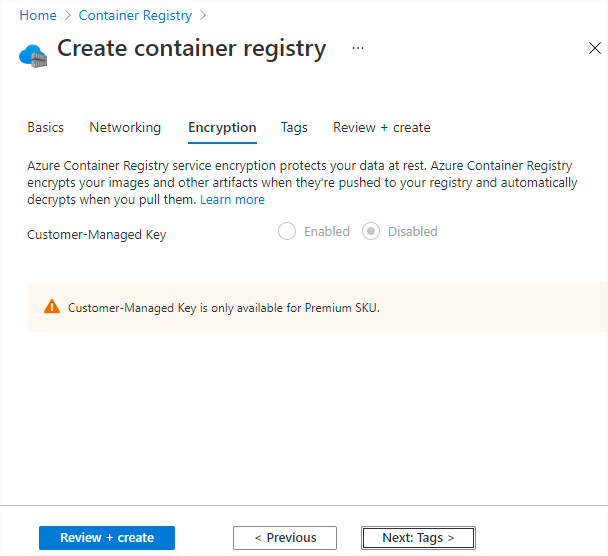
* 1. Ingresamos la siguiente configuración, modificando el name con el numero del grupo que les toco (AzureD4GX) y hacemos clic en Next.



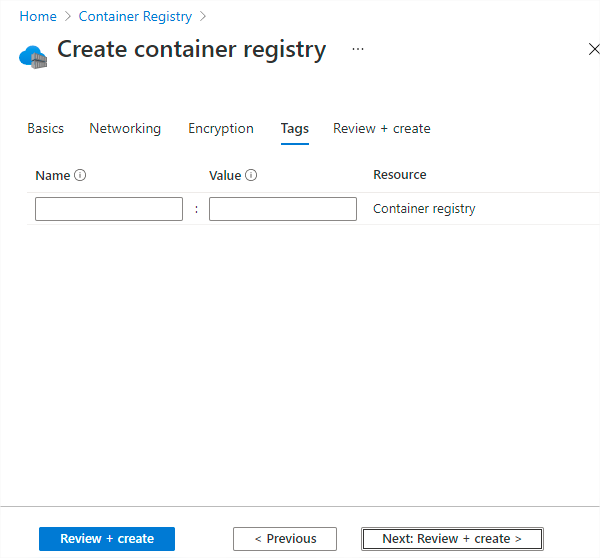
* 1. En la parte de Networking no vamos a tocar nada solo hacemos clic en Next.



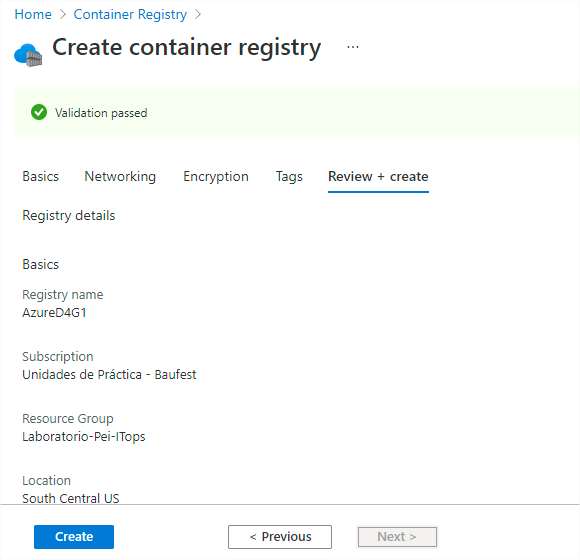
* 1. En la parte de Encryption no vamos a tocar nada solo hacemos clic en Next.



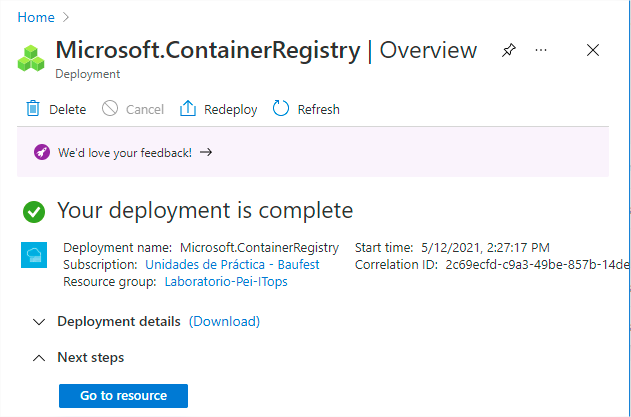
* 1. En la parte de Tags no vamos a tocar nada solo hacemos clic en Next.

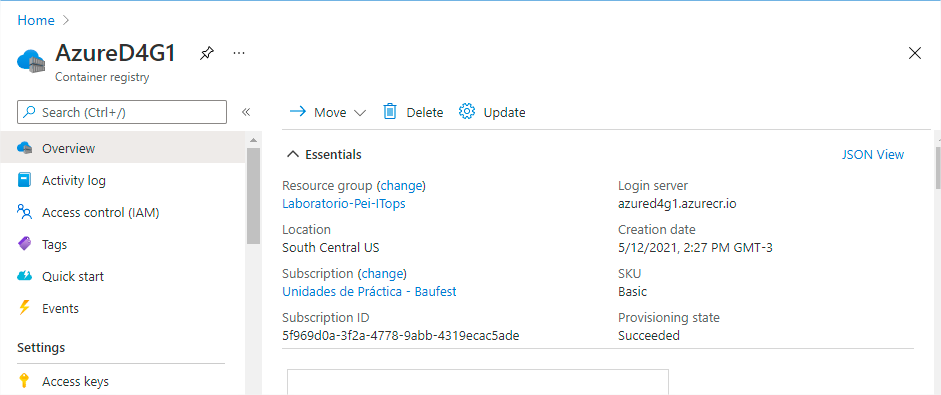


* 1. En la siguiente pantalla validamos que los datos sean correctos y hacemos clic en “Create”.

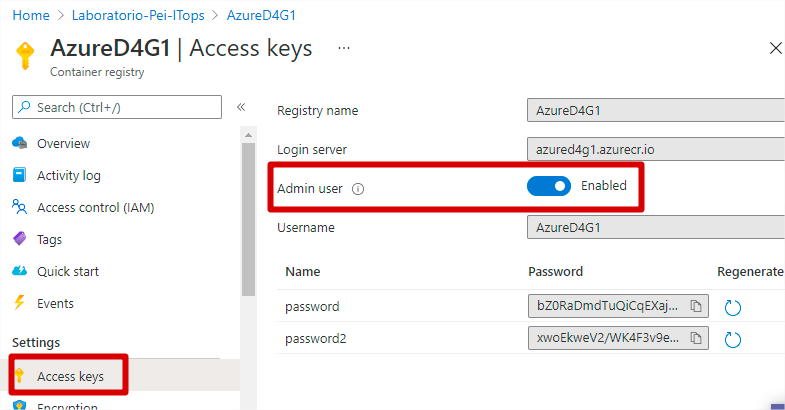


* 1. Una vez terminada la creación hacemos clic en “Go to resource”, para ver el recurso creado:

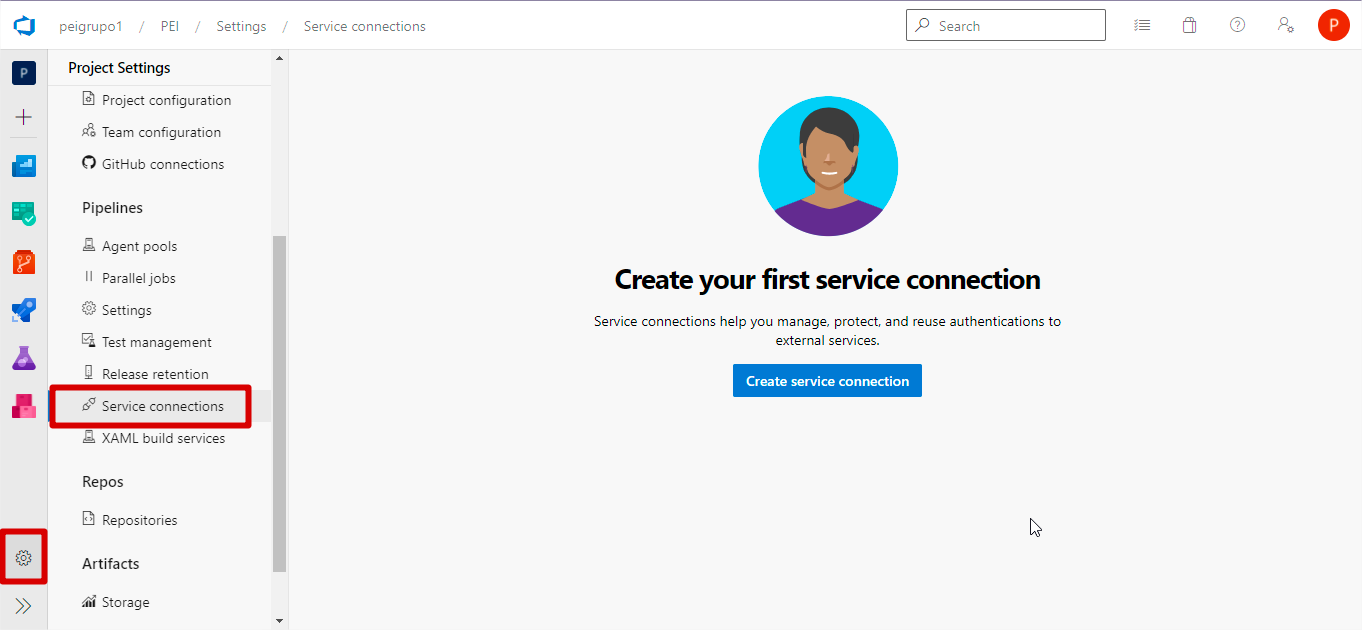




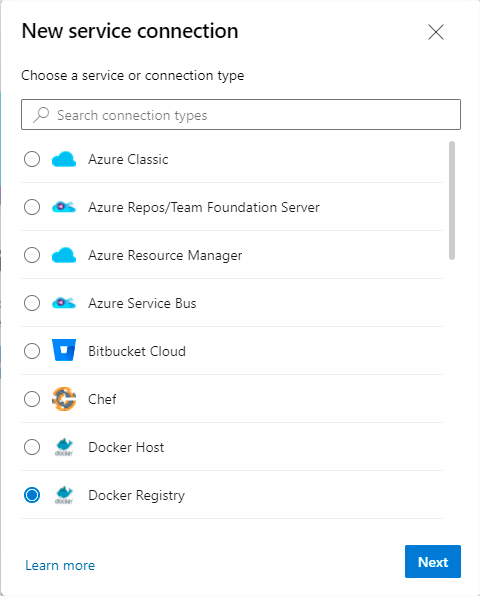
* 1. Dentro del recurso vamos a habilitar el usuario admin, para esto vamos a “Access Keys” y ponemos en Enabled el Admin user



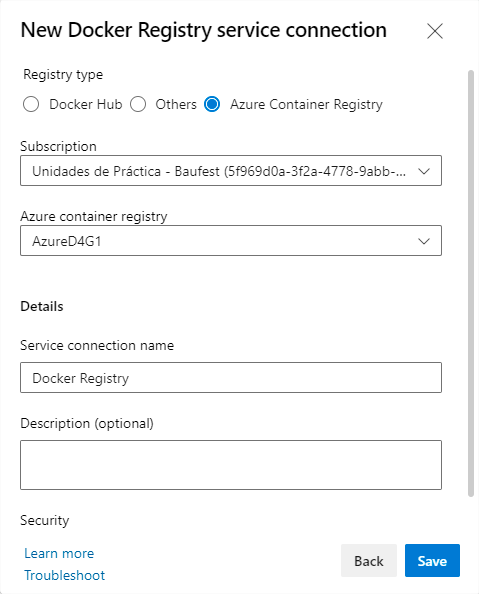
1. Una vez creado el container registry volvemos al Azure Devops (<https://dev.azure.com>) para configurar la conexión con el recurso.
   1. Vamos a la configuración del proyecto y buscamos “service Connections”



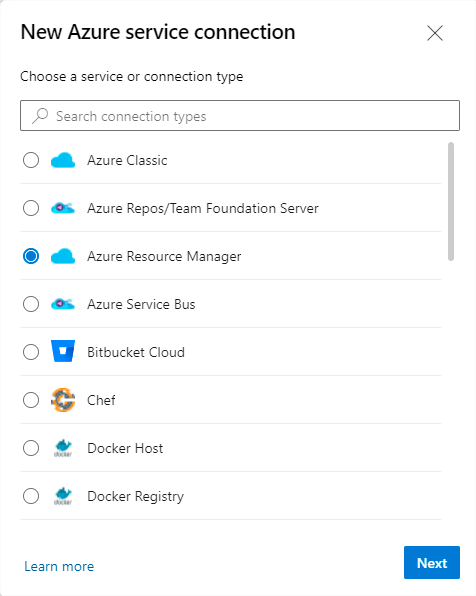
* 1. Hacemos clic en “Create service connection”



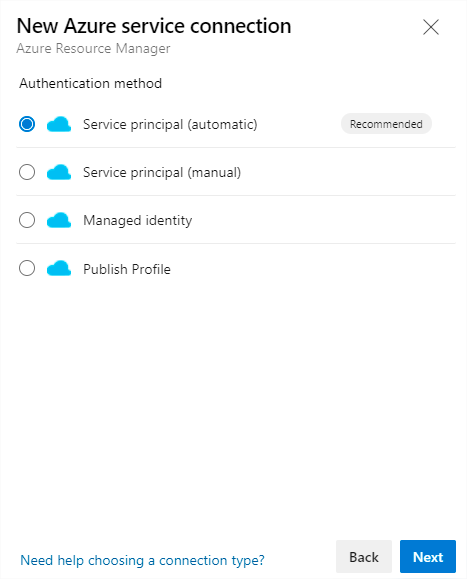
* 1. Seleccionamos la suscripción y el container registry que creamos anteriormente.



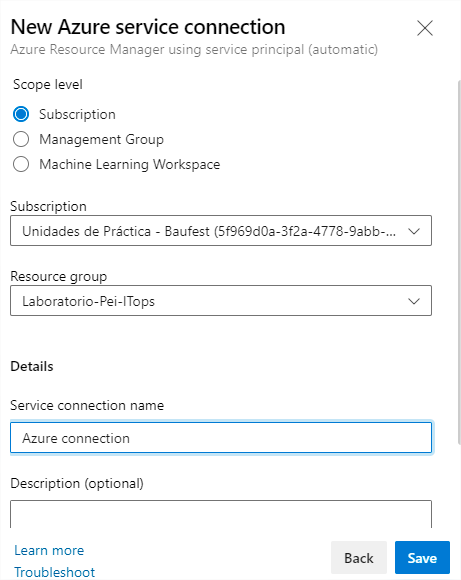
* 1. Creamos un nuevo servicio de Resource Manager



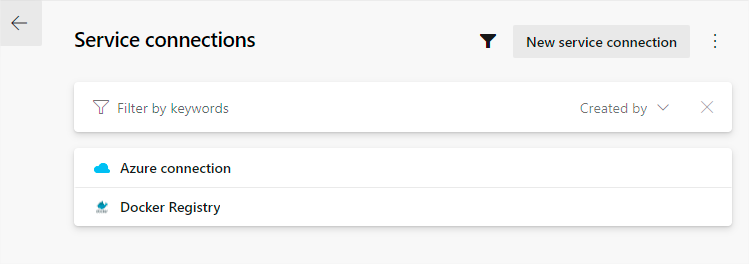
* 1. Elegimos Service principal (automatico)



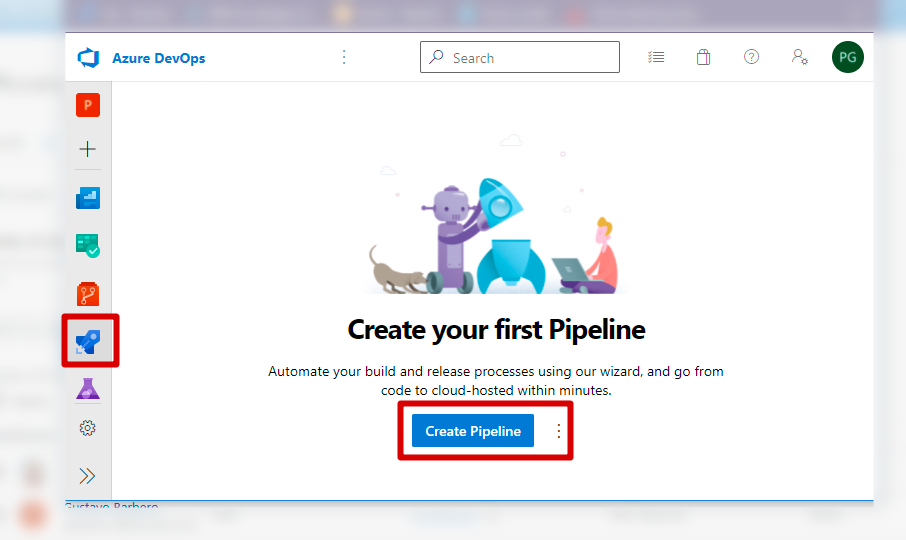
* 1. Completamos los datos como en la imagen y le damos save



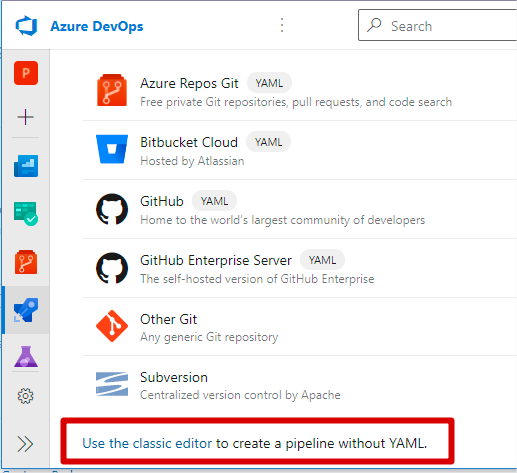
* 1. Podemos ver los dos servicios creados de forma correcta



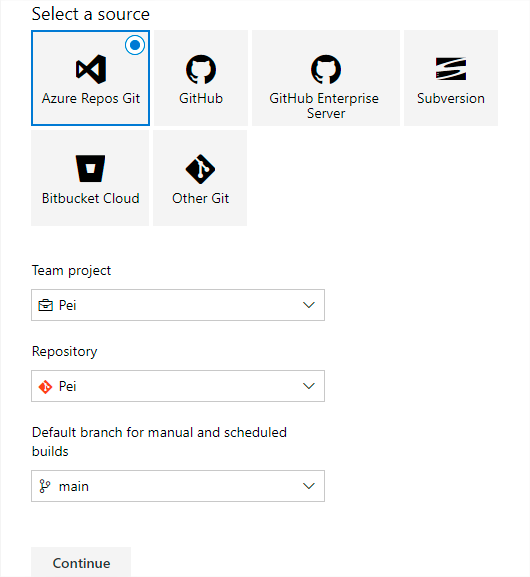
1. Creación del Pipeline
   1. Vamos a la parte de pipelines y hacemos clic en “Create Pipeline”



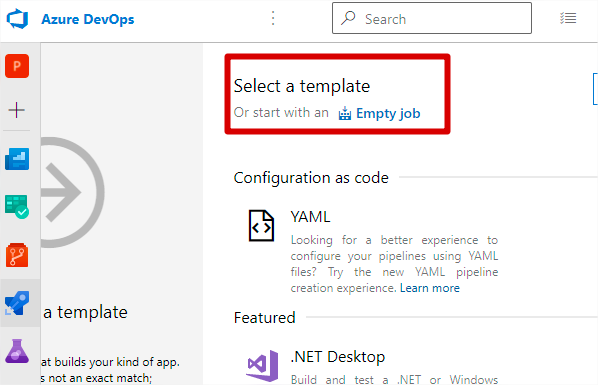
* 1. Elegimos la opción de Classic editor



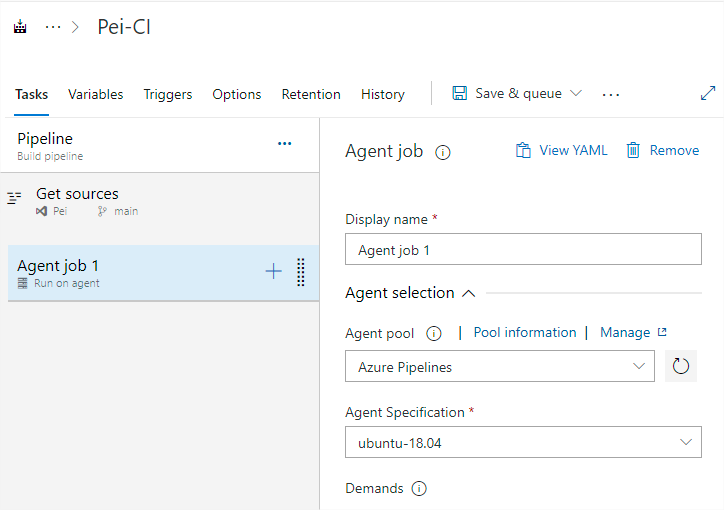
* 1. Seleccionamos el repositorio y hacemos clic en continuar



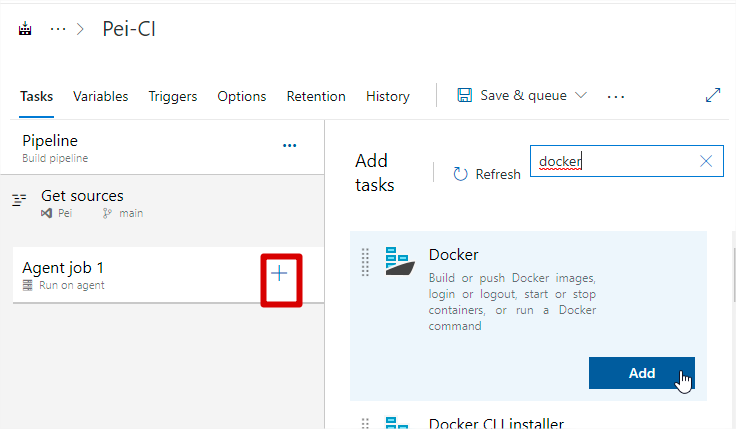
* 1. Elegimos la opción de “Empty job”



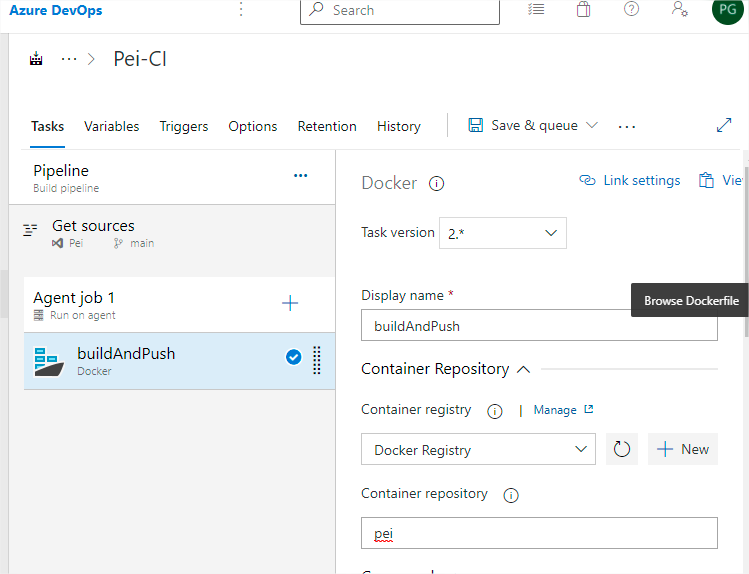
* 1. Esto nos cree el pipeline vacío, donde configuramos el agente que vamos a usar para trabajar, seleccionando “Azure Pipelines” y “Ubuntu-18.04”



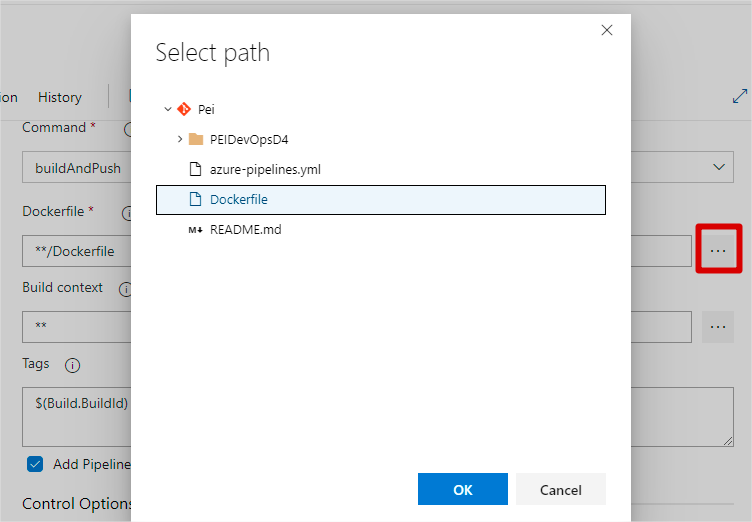
* 1. Hacemos clic en el + y agregamos la tarea de Docker.



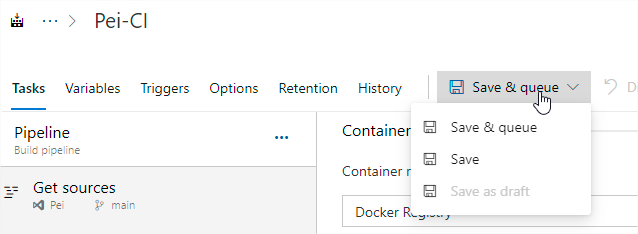
* 1. Completamos los datos, seleccionando la conexión al registry que creamos y un nombre para la imagen que vamos a crear.



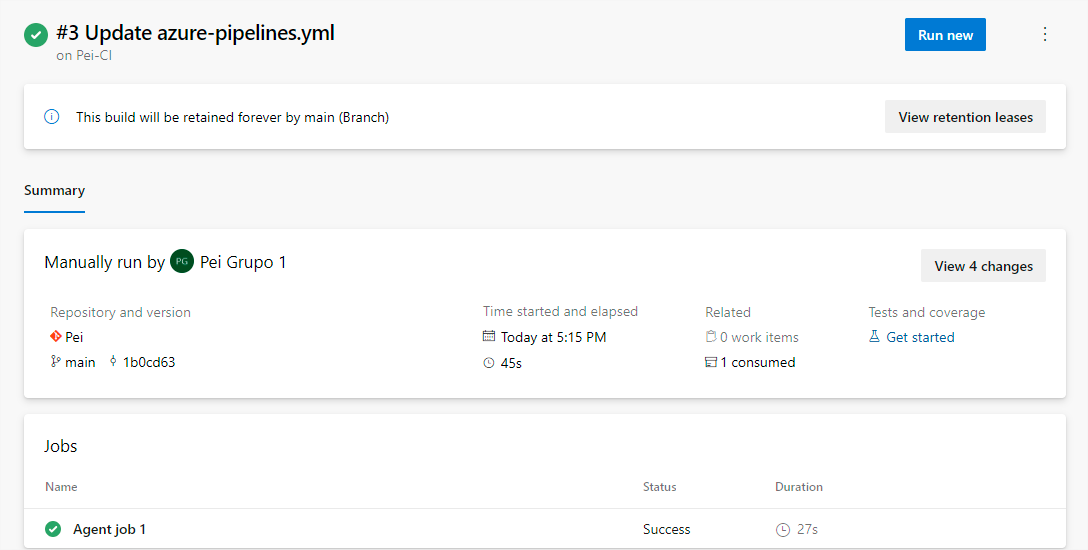
* 1. Seleccionamos el Dockerfile del repositorio y con esto nos queda configurada la tarea de Docker.

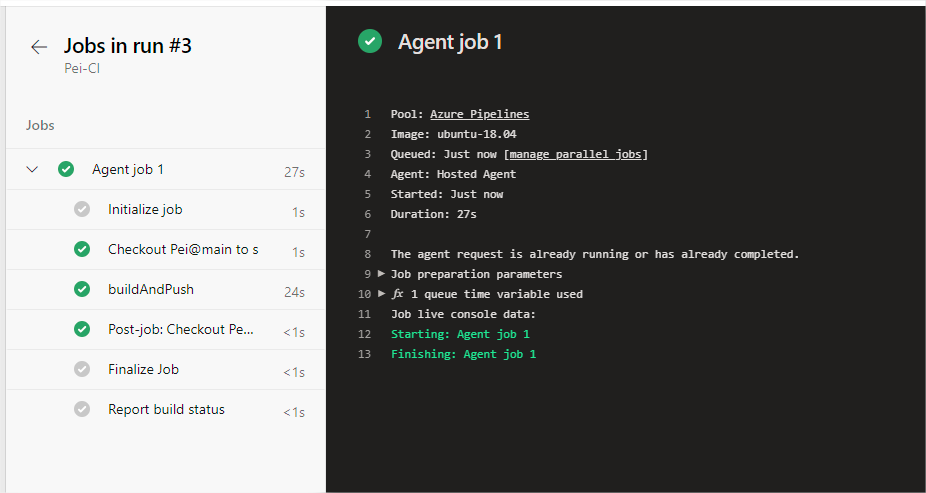


* 1. Hacemos clic en Save & queue, para guardar y ejecutar el pipeline.

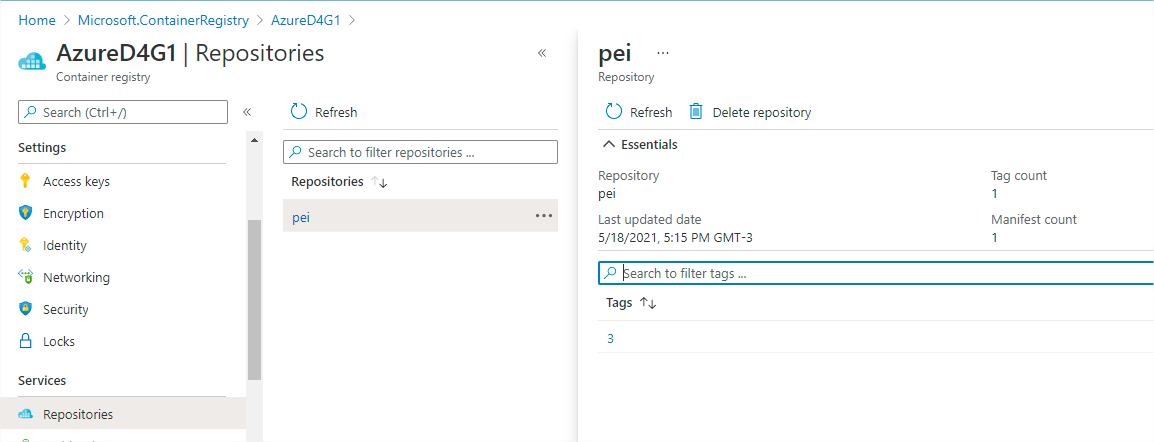


* 1. Si todo se correctamente vamos a ver que el pipeline se ejecuta sin problemas.

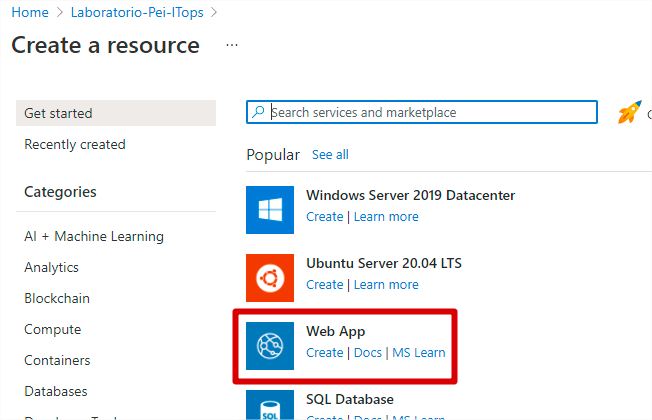




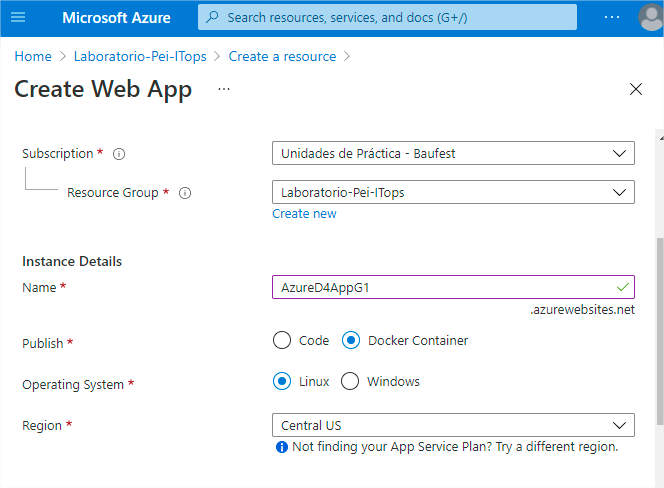
1. Una vez que se ejecuta el pipeline podemos ver en el recurso del container el repositorio de imagen que creamos.



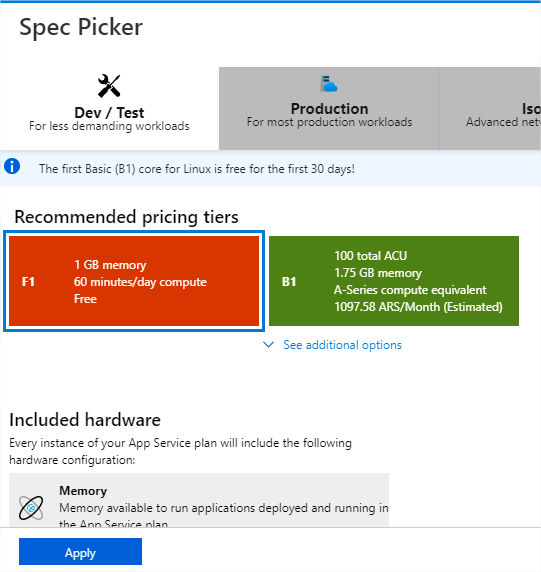
1. El siguiente paso es crear el Web App donde vamos a publicar nuestra página
   1. Buscamos la opción de Web App en la lista de recursos



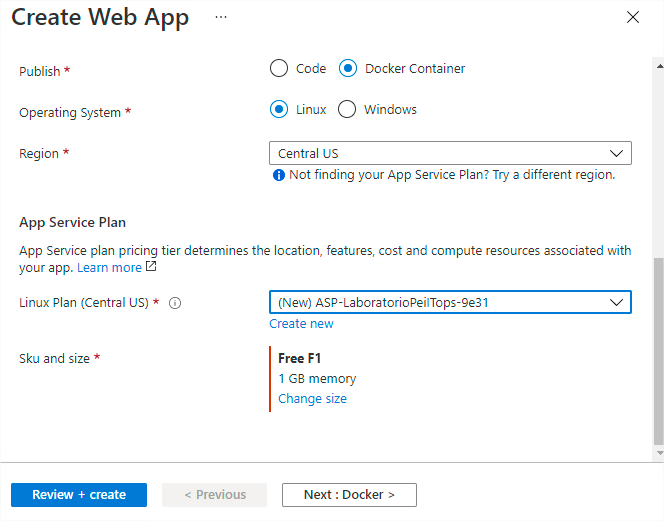
* 1. Elegimos un nombre para el recurso, seleccionamos “Docker container” y “Linux”



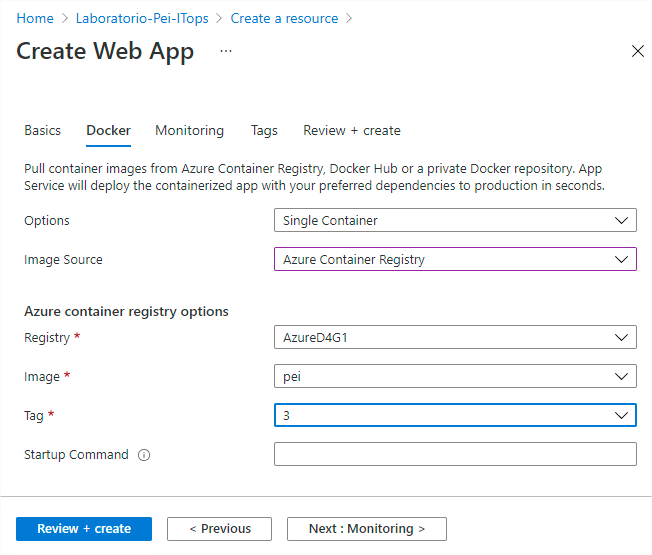
* 1. Modificamos el service plan, por el F1 que es gratis



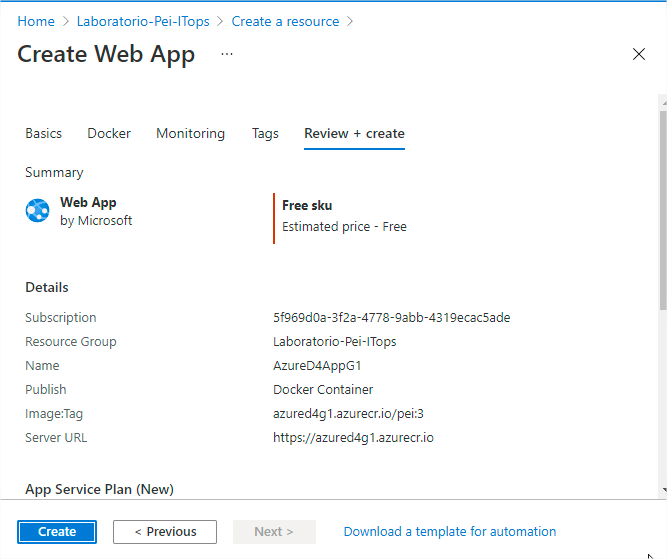
* 1. Nos queda como esta en la siguiente imagen



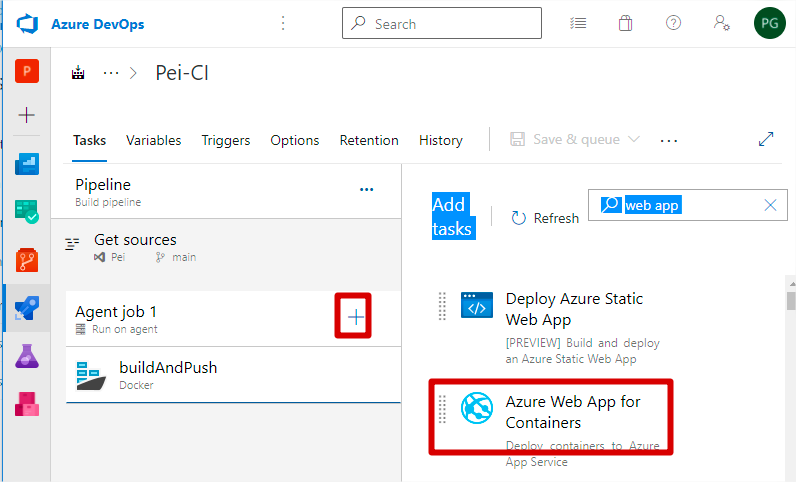
* 1. En la pestaña de Docker seleccionamos el origen de la imagen y dejamos el resto como aparece solo y hacemo clic en “Review + create”



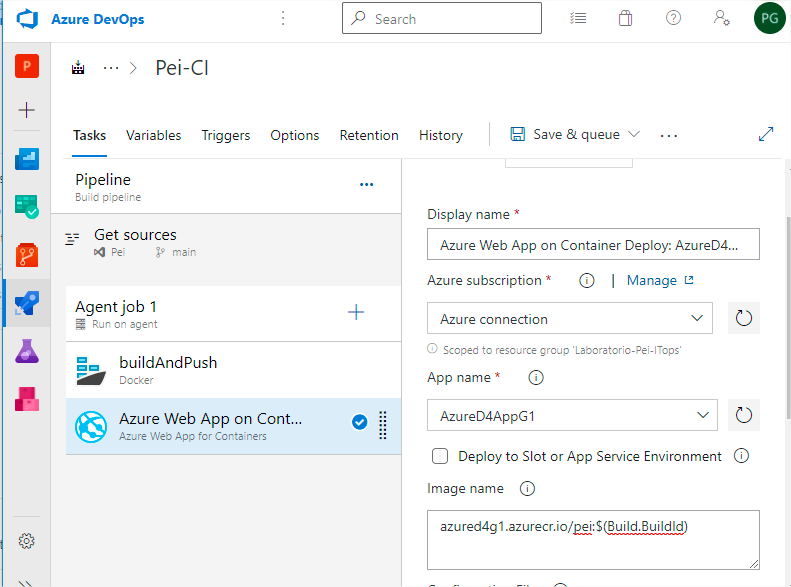
* 1. Hacemos clic en “Create” para crear el recurso.

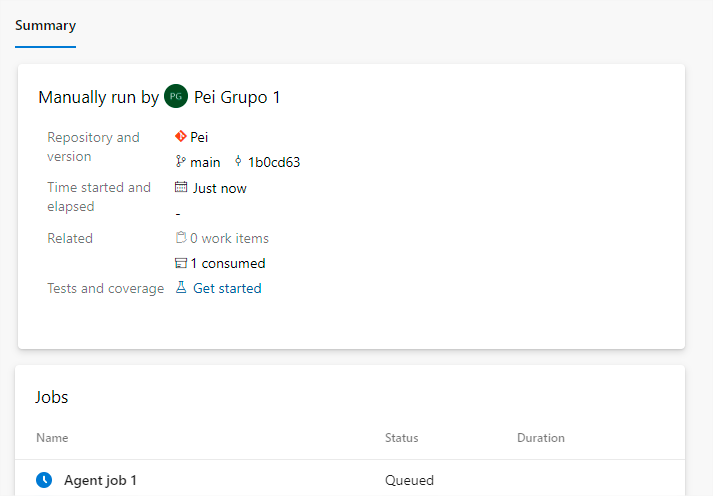


1. Volvemos al Azure Devops y editamos el pipeline que creamos.
   1. Agregamos la tarea para el web app



* 1. Configuramos la tarea con el web app que creamos y el nombre de la imagen y hacemos clic en Save & queue para que corra el pipeline





1. Si todo salió bien podemos abrir la url de nuestra web app y podemos ver la página que desplegamos.

