



Manual de configuraciones

Hugo Jiménez Chévez

DevOps
Versión: 1.0
Octubre, 2023



Contenido

1. Introducción	3
2. Herramientas para utilizar.	3
2.1. Sonarqube.....	3
2.1.1. Dependencias java.....	5
2.2. Docker.....	6
2.3. Azure Agent Pool SelfHosted	9
2.4. Container Registry.....	12
2.5. Kubernetes.....	16
3. Azure DevOps	22
3.1. Pipeline.....	23
3.1.1. Pipeline con tareas de azure.....	23
3.1.2. Pipeline yaml.....	42



1. Introducción

El uso de despliegues en Kubernetes representa un pilar fundamental en la gestión de contenedores y la orquestación de aplicaciones en entornos de infraestructura escalable. A través de este mecanismo, las organizaciones pueden lograr una mayor eficiencia en la implementación y actualización de sus aplicaciones, garantizando la disponibilidad, escalabilidad y fiabilidad deseada. En esta introducción, exploraremos en detalle los conceptos clave, ventajas y prácticas recomendadas para aprovechar al máximo los despliegues de Kubernetes.

2. Herramientas para utilizar.

2.1. Sonarqube

Para utilizar SonarQube, es imprescindible descargar los complementos necesarios que permitirán la puesta en marcha de SonarQube. Estos complementos se encuentran disponibles para su descarga en la página oficial de SonarQube.

<https://www.sonarsource.com/products/sonarqube/downloads/success-download-community-edition/>

En esta ocasión, es necesario descargar la versión "Community" de SonarQube. Una vez descargada, se obtiene un archivo ZIP que debemos ubicar en la carpeta de destino donde deseamos descomprimirlo y nos ubicamos dentro de la carpeta /bin/Windows-86-64. Luego, procedemos a



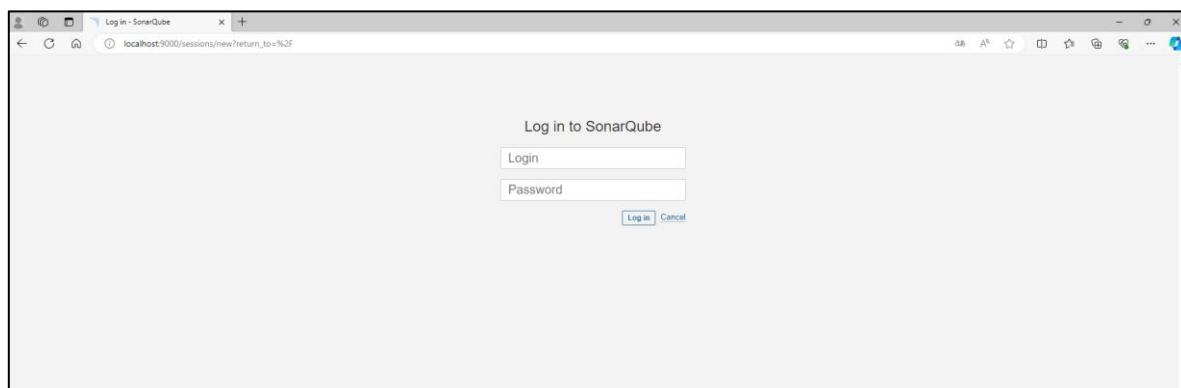
abrir una ventana de comandos (CMD) para ejecutar el siguiente comando.

Ejecutar en cmdshell > StartSonar.bat

A screenshot of a Windows Command Prompt window titled "C:\Windows\System32\cmd.exe - StartSonar.bat". The window shows the execution of a long Java command to start SonarQube. The output includes several warning messages about deprecated methods in java.lang.System and org.sonar.process.PluginSecurityManager, followed by informational messages indicating that the web and compute engine processes are up. The final message states "SonarQube is operational".

```
...pens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/jdk.internal.ref=ALL-UNNAMED --add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.base/sun.nio.ch=ALL-UNNAMED --add-opens=java.management/sun.management=ALL-UNNAMED --add-opens=jdk.management/com.sun.management.internal=ALL-UNNAMED -Dcom.redhat.fips=false -Xmx512m -Xms128m -XX:+HeapDumpOnOutOfMemoryError -Dhttp.nonProxyHosts=localhost|127.*|:::1] -cp ./lib/sonar-application-10.2.1.78527.jar;C:\Users\hugo.jimenez\Sonar\sonarqube-10.2.1.78527\lib\jdbc\h2\h2-2.1.214.jar org.sonar.server.app.WebServer C:\Users\hugo.jimenez\Sonar\sonarqube-10.2.1.78527\temp\sq-process17790445057246081019properties
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.sonar.process.PluginSecurityManager (file:/C:/Users/hugo.jimenez/Sonar/sonarqube-10.2.1.78527/lib/sonar-application-10.2.1.78527.jar)
WARNING: Please consider reporting this to the maintainers of org.sonar.process.PluginSecurityManager
WARNING: System::setSecurityManager will be removed in a future release
2023.10.22 11:11:41 INFO app[[]o.s.a.SchedulerImpl] Process[web] is up
2023.10.22 11:11:41 INFO app[[]o.s.a.ProcessLauncherImpl] Launch process[COMPUTE_ENGINE] from [C:\Users\hugo.jimenez\Sonar\sonarqube-10.2.1.78527\temp\sq-process1554629364311065121properties
WARNING: A terminally deprecated method in java.lang.System has been called
WARNING: System::setSecurityManager has been called by org.sonar.process.PluginSecurityManager (file:/C:/Users/hugo.jimenez/Sonar/sonarqube-10.2.1.78527/lib/sonar-application-10.2.1.78527.jar)
WARNING: Please consider reporting this to the maintainers of org.sonar.process.PluginSecurityManager
WARNING: System::setSecurityManager will be removed in a future release
2023.10.22 11:11:44 INFO app[[]o.s.a.SchedulerImpl] Process[ce] is up
2023.10.22 11:11:44 INFO app[[]o.s.a.SchedulerImpl] SonarQube is operational
```

El cual nos ejecutarán los procesos y una vez que indique que el sonar está en ejecución procedemos a levantarlo



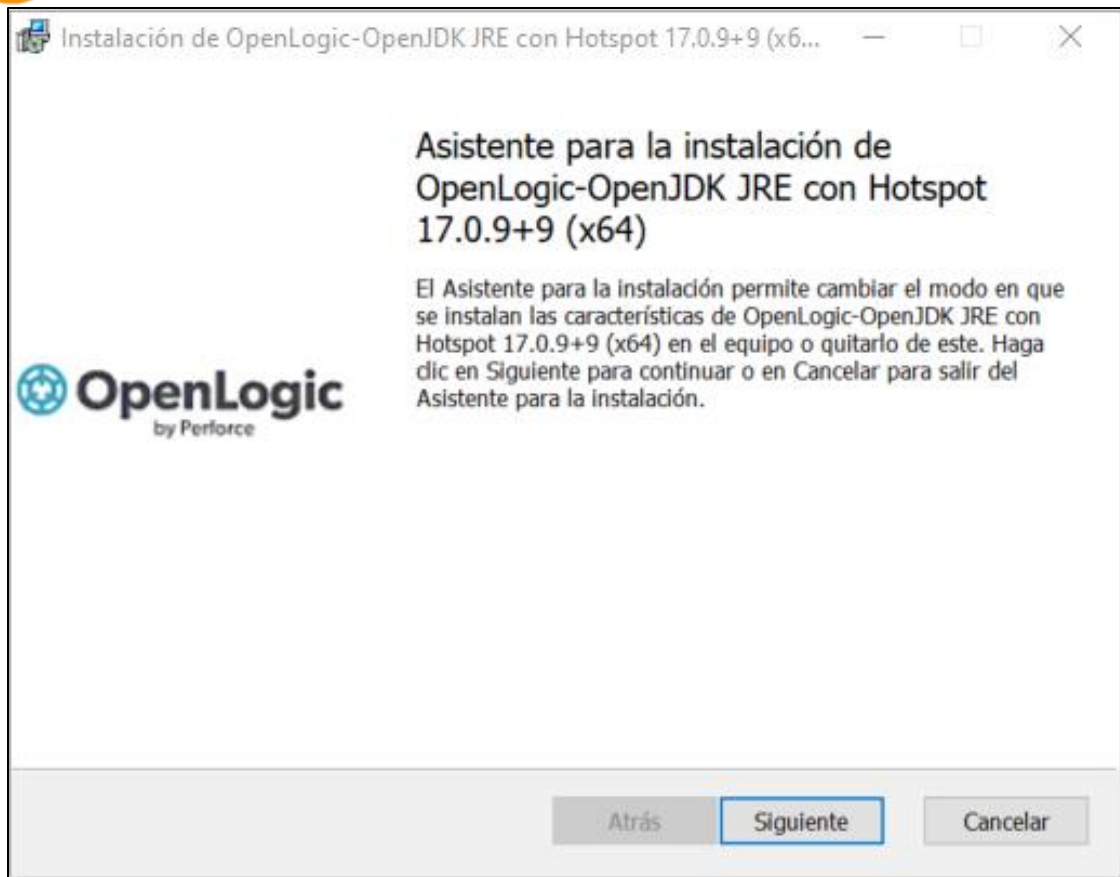


2.1.1. Dependencias java

Es importante mencionar que para que la aplicación de Java funcione es necesario instalar el complemento de java el cual se obtiene del siguiente enlace de open logic.

https://www.openlogic.com/openjdk-downloads?field_java_parent_version_target_id=807&field_operating_system_target_id=436&field_architecture_target_id=All&field_java_package_target_id=401

El cual nos descargara un ejecutable y solo debemos dar clic sobre el y nos levantara la siguiente pantalla en el cual la instalación es muy sencilla, ya que solo es necesario darle en siguiente en todos los pasos y se instalara sin problemas.



2.2. Docker

Para aprovechar Docker al máximo, es esencial obtener los componentes necesarios que habilitarán la implementación de SonarQube. Estos componentes se encuentran disponibles para su descarga en el sitio web oficial de Docker.

<https://docs.docker.com/desktop/install/windows-install/>

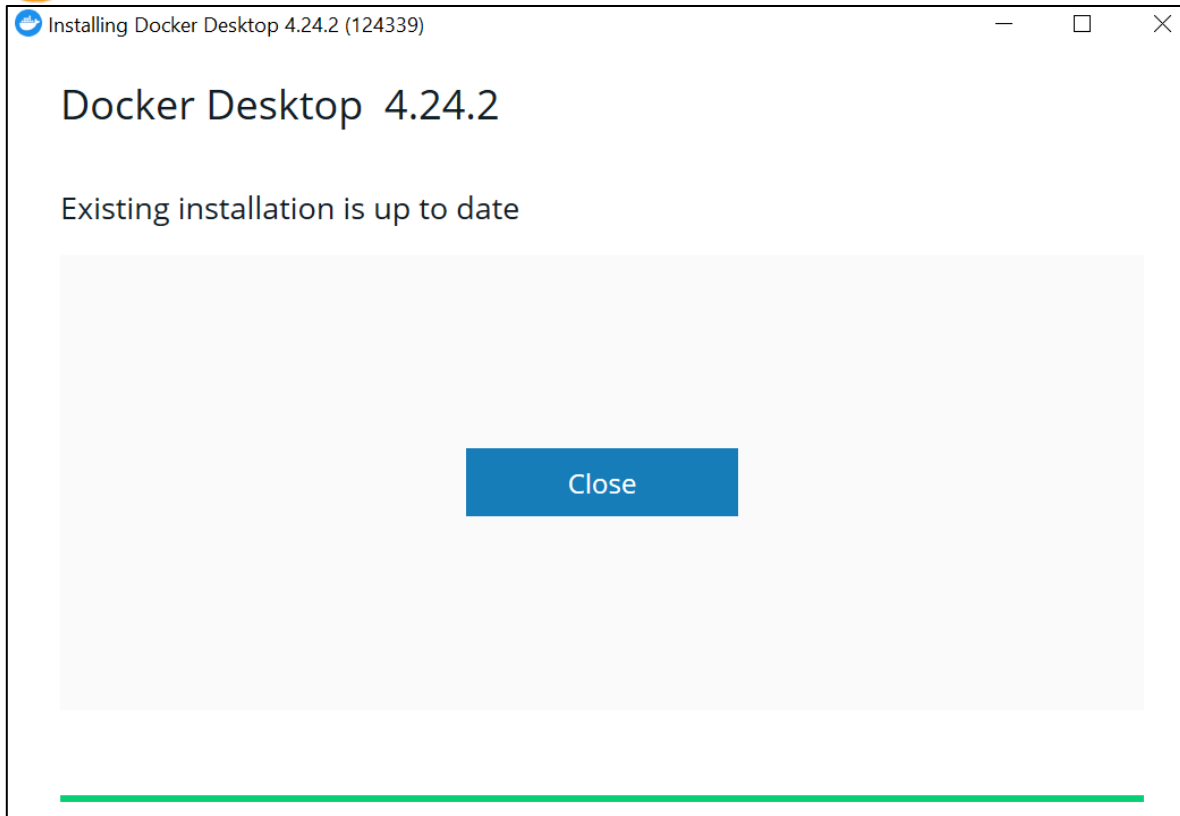


Es importante mencionar que primero debemos tener una cuenta en Docker hub para instalar la versión de Docker de escritorio, ya que es esta es necesaria para las compilaciones de más adelante.

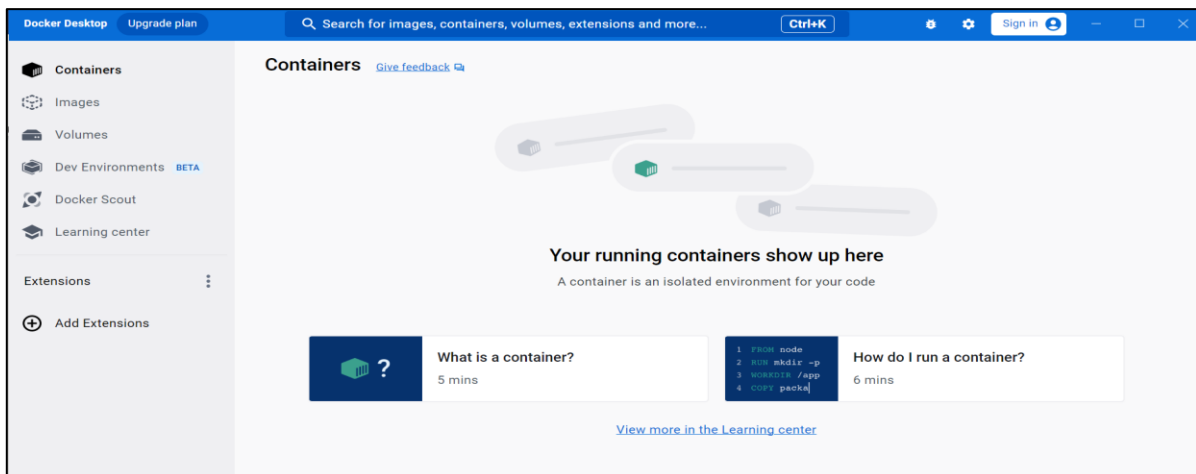
Dicho lo anterior, ubicamos el ejecutable que descargamos del enlace anterior y solo debemos dar clic y nos saldrá la siguiente pantalla, la cual debemos mantener ambas opciones que nos aparecen marcadas y le damos "ok".



Una vez que le hayamos dado en "ok", comenzara la instalación y solo debemos esperar que termine y le damos en "close".



Y al levantar la aplicación nos debería salir de la siguiente forma.



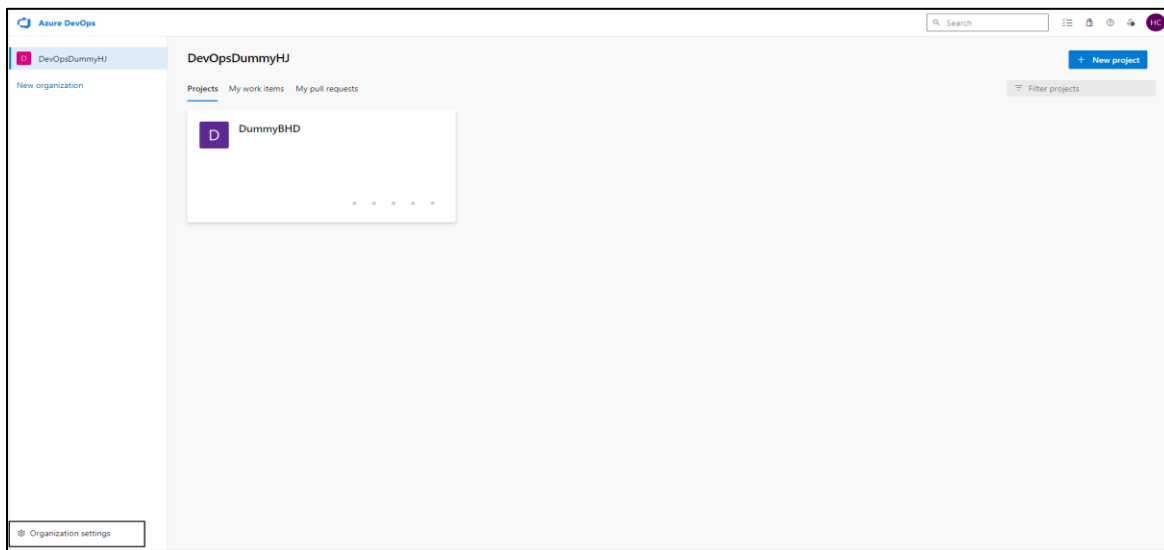


2.3. Azure Agent Pool SelfHosted

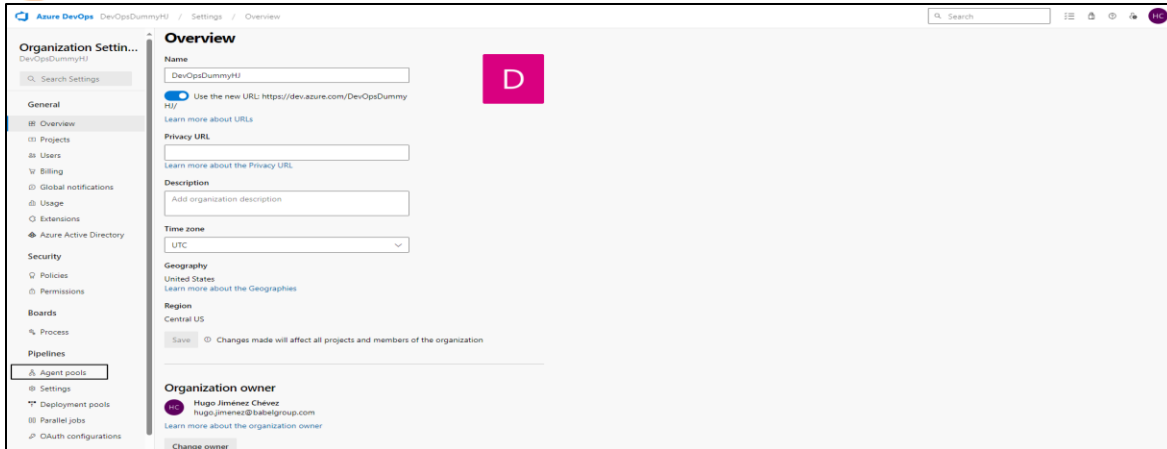
Para Instalar el azure agent pool para las compilaciones, es crucial adquirir los componentes necesarios que permitirán la implementación de este agente. Estos componentes están disponibles para su descarga en Azure DevOps.

Por lo que debemos seguir los siguientes pasos.

Primeramente, se debe ingresar en azure DevOps en la sección de "Organization Setting".



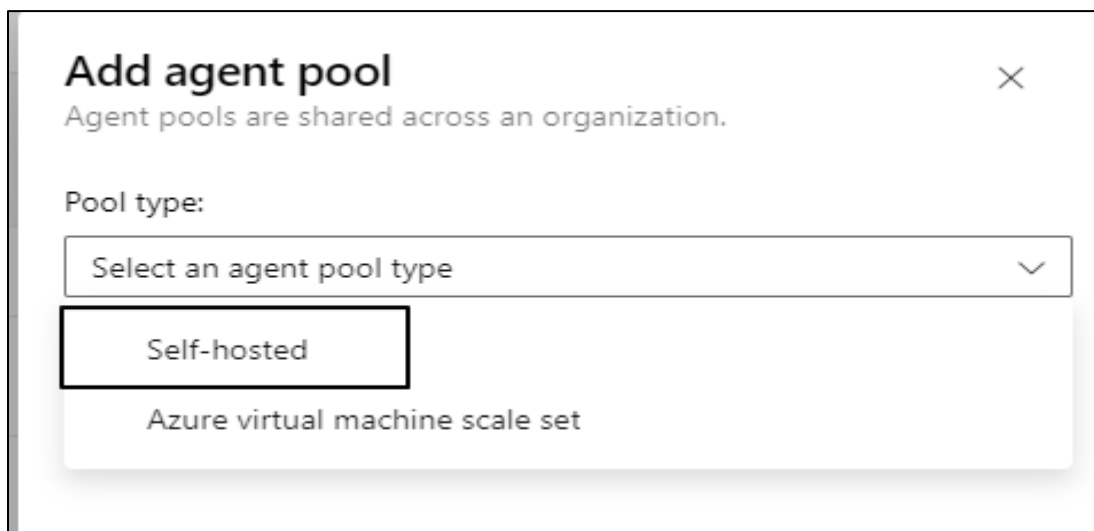
Una vez ingresado, procedemos a ingresar en la sección de "Agent pool"



Luego, nos dirigimos a agregar nuevo pool "Add pool"



En la pantalla que nos saldrá después de dar clic en agregar nuevo pool, debemos seleccionar la opción de "Self hosted"





Una vez hayamos elegido la opción anterior debemos llenar los siguientes datos.

Add agent pool

Agent pools are shared across an organization.

Pool type:

Self-hosted

A pool of agents that you set up and manage on your own to run jobs. [Learn more.](#)

Name:

New agent pool name

Description (optional):

Markdown supported.

Pipeline permissions:

☒ Auto-provision this agent pool in all projects

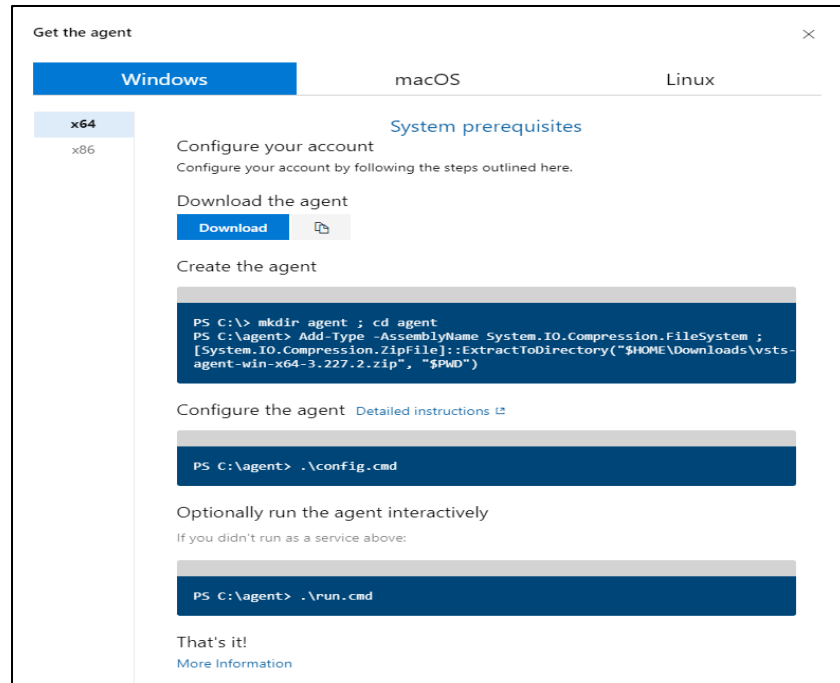
Posteriormente, seleccionamos el agente que creamos



Y luego seleccionamos la opción "New agent"



El cual nos desplegara la siguiente pantalla con toda la información necesaria.



Una vez que hayamos hecho todos los pasos anteriores, se creará un nuevo servicio windows en el cual debemos asignar un usuario con permisos administradores para acceder a todos los complementos necesarios para las compilaciones.

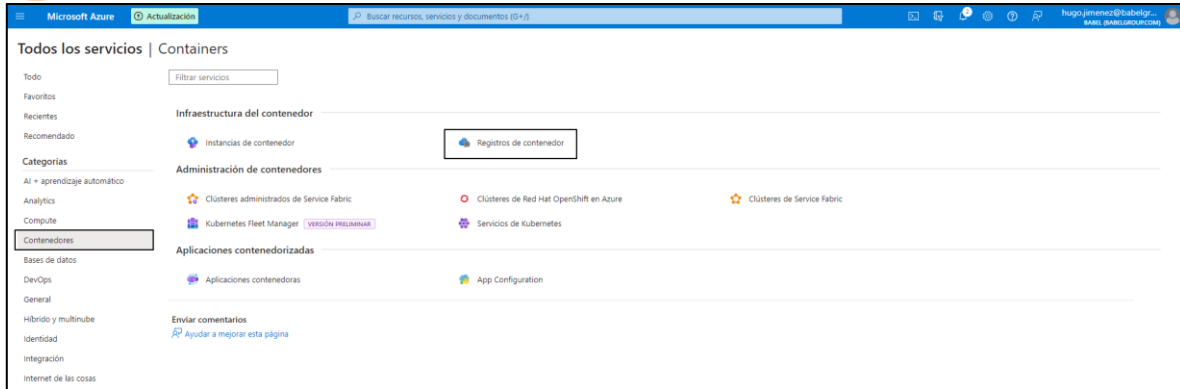


2.4. Container Registry

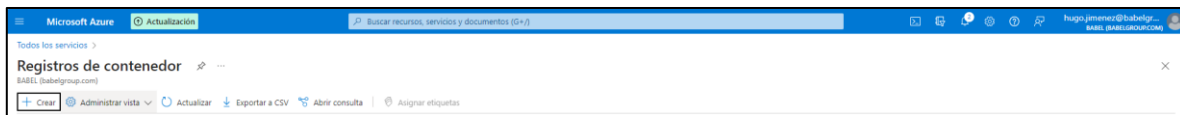
Para configurar nuestro contenedor donde almacenaremos nuestras imágenes Docker, debemos hacer los siguientes pasos, primeramente, ingresamos al portal del azure.

<https://portal.azure.com/>

Posteriormente, buscamos todos los servicios y nos dirigimos en la sección de contenedores y elegimos la opción de "Registros de contenedores".



Una vez hayamos ingresado en los contenedores, procedemos a crear uno nuevo.



Una vez que haga clic en "Crear", se abrirá la siguiente pantalla, donde se le solicitará la información del contenedor. Debe proporcionar el nombre del grupo de recursos (es fundamental utilizar el mismo para esta configuración), asignar un nombre al contenedor, seleccionar una ubicación geográfica y elegir el plan de precios estándar.



Microsoft Azure Actualización Buscar recursos, servicios y documentos (G+/I)

Todos los servicios > Registros de contenedor >

Crear Registro de contenedor

Datos básicos Redes Cifrado Etiquetas Revisar y crear

Azure Container Registry permite compilar, almacenar y administrar artefactos e imágenes de contenedor en un registro privado para todos los tipos de implementación de contenedor. Use registros de contenedor de Azure con sus canalizaciones de desarrollo e implementación de contenedores actuales. Use Azure Container Registry Tasks para compilar imágenes de contenedor en Azure a petición, o bien automatizar compilaciones desencadenadas por actualizaciones del código fuente, actualizaciones de la imagen base de un contenedor o temporizadores. [Más información](#)

Detalles del proyecto

Suscripción * Azure subscription 1

Grupo de recursos * rg-dummy-bhd [Crear nuevo](#)

Detalles de instancia

Nombre del Registro * Azcontainerbhd [azurecr.io](#)

Ubicación * East US

Uso de zonas de disponibilidad ☐

Las zonas de disponibilidad se activan en registros premium y en regiones que admiten zonas de disponibilidad. [Más información](#)

Plan de precios * Estándar

En la siguiente sección dejamos los datos por defectos y continuamos.

Microsoft Azure Actualización Buscar recursos, servicios y documentos (G+/I)

Todos los servicios > Registros de contenedor >

Crear Registro de contenedor

Datos básicos **Redes** Cifrado Etiquetas Revisar y crear

Conectividad de red

Puede conectarse a este registro de forma pública mediante direcciones IP públicas, o bien de forma privada con un punto de conexión privado. [Más información](#)

Configuración de conectividad

☒ Acceso público (a todas las redes)

☐ Acceso privado (recomendado)

El acceso privado (recomendado) solo está disponible para los planes de precios Premium.

En esta pantalla de cifrado, dejamos todo por defecto.

Microsoft Azure Actualización Buscar recursos, servicios y documentos (G+/I)

Todos los servicios > Registros de contenedor >

Crear Registro de contenedor

Datos básicos Redes **Cifrado** Etiquetas Revisar y crear

El cifrado del servicio Azure Container Registry protege los datos en reposo. Azure Container Registry cifra las imágenes y otros artefactos cuando se insertan en el registro y los descifra automáticamente al extraerlos. [Más información](#)

Clave administrada por el cliente ☐ Habilitado ☒ Deshabilitado

La clave administrada por el cliente solo está disponible para la SKU Premium.



Con respecto a las etiquetas, pueden omitir esta parte, pero es importante asignarle algunos para referencias a que proyecto pertenece, quien lo creo o la descripción del recurso creado.

Todos los servicios > Registros de contenedor >

Crear Registro de contenedor

Datos básicos Redes Cifrado **Etiquetas** Revisar y crear

Nombre ⓘ	Valor ⓘ	Recurso
<input type="text"/>	<input type="text"/>	Registro de contenedor

Como paso final, solo validamos que los datos estén correctos y procedemos a crear el “registro de contenedor”

Microsoft Azure Actualización Buscar recursos, servicios y documentos (G+)

Todos los servicios > Registros de contenedor >

Crear Registro de contenedor

✓ Validación superada

Datos básicos Redes Cifrado Etiquetas **Revisar y crear**

Detalles del registro

Datos básicos	
Nombre del Registro	Azcontainerbhd
Suscripción	Azure subscription 1
Grupo de recursos	rg-dummy-bhd
Ubicación	East US
Zonas de disponibilidad	Deshabilitado
Plan de precios	Standard
Redes	
Acceso de red pública	Sí
Cifrado	
Clave administrada por el cliente	Deshabilitado
Identidad	Ninguno
Key Vault	Ninguno
Clave de cifrado	Ninguno
Versión	Ninguno

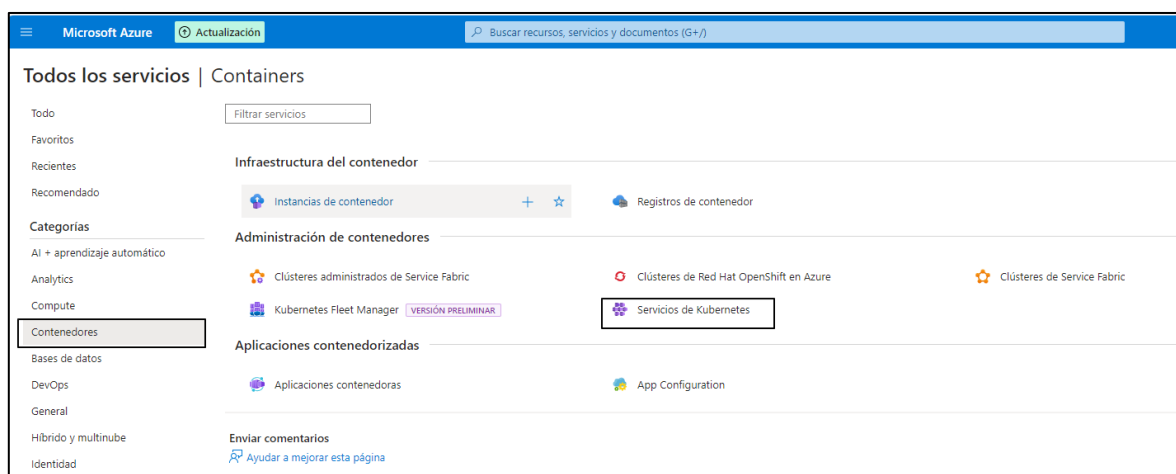


2.5. Kubernetes

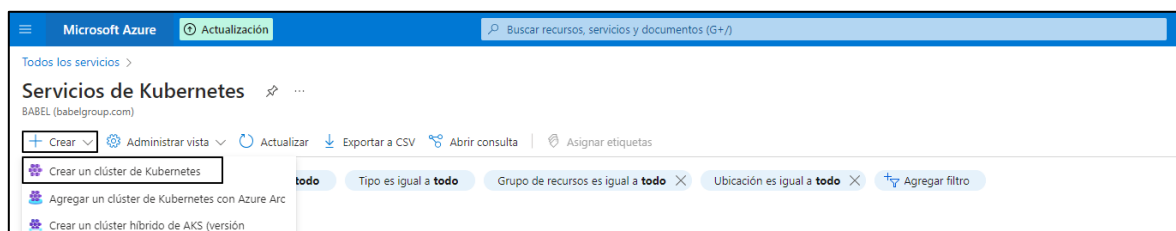
Para configurar nuestro Kubernetes donde despleguemos nuestras imágenes almacenadas en el contenedor, debemos hacer los siguientes pasos, primeramente, ingresamos al portal del azure.

<https://portal.azure.com/>

Posteriormente, buscamos todos los servicios y nos dirigimos en la sección de contenedores y elegimos la opción de “Servicios de Kubernetes”.



Una vez hayamos ingresado en los servicios de Kubernetes, procedemos a crear uno nuevo y elegimos la opción para esta prueba “Crear un clúster de Kubernetes”.



Una vez que haga clic en "Crear", se abrirá la siguiente pantalla, donde se le solicitará la información del Kubernetes. Debe proporcionar el nombre del



grupo de recursos (es fundamental utilizar el mismo para esta configuración), asignar la configuración preestablecida del clúster, asignar un nombre al Kubernetes, seleccionar una ubicación geográfica, zonas de disponibilidad, plan de tarifa (en este caso se usó el estándar) y por último la autenticación.

Microsoft Azure Actualización

Todos los servicios > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Datos básicos Grupos de nodos Redes Integraciones Opciones avanzadas Etiquetas Revisar y crear

Azure Kubernetes Service (AKS) administra el entorno de Kubernetes hospedado, a la vez que facilita y agiliza la implementación y la administración de aplicaciones en contenedores sin necesidad de experiencia relativa. También elimina la carga de las operaciones en curso y el mantenimiento mediante el aprovisionamiento, la actualización y el escalado de los recursos a petición, sin tener que desconectar las aplicaciones. [Más información](#)

Detalles del proyecto

Seleccione una suscripción para administrar los recursos implementados y los costos. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción * Azure subscription 1

Grupo de recursos * rg-dummy-bhd [Crear nuevo](#)

Detalles del clúster

Configuración preestablecida del clúster Desarrollo/pruebas

Para personalizar rápidamente el clúster de Kubernetes, elija una de las configuraciones preestablecidas anteriores. Puede modificar estas configuraciones en cualquier momento. [Obtener más información y comparar valores preestablecidos](#)

Nombre del clúster de Kubernetes * kube-bhd

Región * (US) East US

Zonas de disponibilidad Ninguno

Plan de tarifa de AKS Estándar

Versión de Kubernetes * 1.26.6 (predeterminado)

Actualización automática Habilitado con revisión (recomendado)

Elija entre cuentas locales o Azure AD para la autenticación y Azure RBAC o Kubernetes para sus necesidades de autorización.

Autenticación y autorización Cuentas locales con RBAC de Kubernetes

Una vez implementado el clúster, use la CLI de Kubernetes para administrar las configuraciones de RBAC. [Obtener más información](#)

[Anterior](#) [Siguiente: Grupos de nodos](#) [Revisar y crear](#)

En la siguiente sección se deja los valores por defecto.

Microsoft Azure Actualización

Todos los servicios > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Datos básicos Grupos de nodos Redes Integraciones Opciones avanzadas Etiquetas Revisar y crear

Grupos de nodos

Además del grupo de nodos principal necesario configurado en la pestaña Conceptos básicos, también puede agregar grupos de nodos opcionales para controlar una gran variedad de cargas de trabajo. [Más información sobre los grupos de nodos](#)

+ Agregar grupo de nodos Eliminar

Nombre	Modo	Tamaño del nodo	SKU DEL SISTEM...	Número de r
agentpool	Sistema	Standard_DS2_v2 (cambiar)	Ubuntu	2 - 5

Habilitar nodos virtuales

Los nodos virtuales permiten el escalado flexible respaldado por Azure Container Instances sin servidor. [Más información sobre los nodos virtuales](#)

Habilitar nodos virtuales ☐

Cifrado de disco de SO del grupo de nodos

De forma predeterminada, todos los discos de AKS están cifrados en reposo con las claves administradas por Microsoft. Para obtener más control sobre el cifrado, puede proporcionar sus propias claves con un conjunto de cifrado de disco respaldado por Azure Key Vault. El conjunto de cifrado de disco se usará para cifrar los discos del SO de todos los grupos de nodos del clúster. [Más información](#)

Tipo de cifrado (Predeterminado) Cifrado en reposo con una clave administrada por la plata...

[Anterior](#) [Siguiente: Redes](#) [Revisar y crear](#)



En la siguiente pantalla en redes, se utilizó la conexión establecida por defecto.

Microsoft Azure Actualización Buscar recursos, servicios y documentos (2+)

Todos los servicios > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Datos básicos Grupos de nodos **Redes** Integraciones Opciones avanzadas Etiquetas Revisar y crear

Azure proporciona varios controles de red para ayudar a administrar y proteger el acceso al clúster de Kubernetes.

Acceso privado
Habilite un clúster privado para restringir el acceso del nodo de trabajo a la API, lo que mejora la seguridad y el aislamiento de la carga de trabajo de Kubernetes.

Habilitar clúster privado ☐

Acceso público
Establecer intervalos IP autorizados ☐

Redes de contenedores

Conexión de red ☒ **kubernetes**
Ideal para grupos de nodos más pequeños. A cada pod se le asigna una dirección IP lógicamente diferente de la subred para simplificar la configuración.

☐ **Azure CNI**
Ideal para grupos de nodos más grandes. A cada nodo y pod se le asigna una dirección IP única para las configuraciones avanzadas.

Traiga su propia red virtual ☐

Prefijo de nombre DNS *

Directiva de red ☐ Ninguno
Permitir todo el tráfico de entrada y salida a los pods

☒ **Calico**
Solución de redes de código abierto. Ideal para implementaciones a gran escala con requisitos de seguridad estrictos.

☐ **Azure**
Solución de red nativa. Ideal para implementaciones más sencillas con requisitos básicos de seguridad y red.

Se recomienda la directiva de red de Calico para la configuración de Desarrollo/pruebas.

La directiva de red de Azure no es compatible con las redes de kubernetes.

Equilibrador de carga

< Anterior Siguiente: Integraciones > **Revisar y crear**

En la siguiente sección de integraciones debemos asignar el contenedor que creamos anteriormente, de esta forma nos será más fácil realizar los despliegues de las imágenes.



Microsoft Azure Actualización

Buscar recursos, servicios y documentos (G+/I)

Todos los servicios > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Conecte el clúster de AKS con servicios adicionales.

Microsoft Defender for Cloud
Microsoft Defender for Cloud proporciona características unificadas de administración para la seguridad y protección contra amenazas en todas las cargas de trabajo en la nube híbrida. [Más información](#)

✓ Su suscripción está protegida por el plan básico de Microsoft Defender for Cloud.

Azure Container Registry
Conecte el clúster a una instancia de Azure Container Registry para permitir que las implementaciones se realicen sin problemas desde un registro de imagen privada. [Más información sobre Azure Container Registry](#)

Registro de contenedor: [Crear](#)

Azure Monitor
Además de las métricas de la CPU y la memoria incluidas en AKS de forma predeterminada, puede habilitar la información de contenedores para obtener datos más completos sobre el rendimiento y el estado globales del clúster. La facturación se basa en las opciones de configuración de la ingesta y la retención de datos. [Más información acerca de la supervisión del rendimiento y del estado de contenedores](#)
[Obtenga más información sobre precios](#)

☒ Configuración predeterminada
Habilite los registros de contenedor, las métricas de Prometheus y las visualizaciones de Grafana.

☐ Configuración personalizada

☐ Desactivado

Opciones de configuración predeterminadas

Registros de contenedor	
Área de trabajo de Log Analytics	Ninguna

Métricas de Prometheus	
Área de trabajo de Azure Monitor	Ninguna

Visualizaciones de Grafana	
Área de trabajo de Grafana	Ninguna

Alertas

< Anterior Siguiendo: Opciones avanzadas > [Revisar y crear](#)

En opciones avanzadas dejamos por defecto el grupo de recursos que se crear.

Microsoft Azure Actualización

Buscar recursos, servicios y documentos (G+/I)

Todos los servicios > Servicios de Kubernetes >

Crear un clúster de Kubernetes

Datos básicos Grupos de nodos Redes Integraciones Opciones avanzadas Etiquetas Revisar y crear

Habilitar el controlador CSI del almacén de secretos ☐

Grupo de recursos de infraestructura [Editar](#)



En cuanto a las etiquetas, es posible pasar por alto esta sección, sin embargo, resulta crucial asignar algunas para tener referencias sobre a qué proyecto pertenece, quién lo creó o la descripción del recurso creado.

The screenshot shows the 'Etiquetas' (Tags) tab in the 'Crear un clúster de Kubernetes' (Create a Kubernetes cluster) wizard. The page includes a search bar at the top and a navigation menu. The main content area has a tabbed interface with 'Etiquetas' selected. Below the tabs, there is a text box explaining that tags are used to categorize resources and view consolidated billing. A note mentions that tags are updated automatically when resources are created or modified. At the bottom, there are input fields for 'Nombre' (Name), 'Valor' (Value), and a 'Recurso' (Resource) dropdown menu showing '2 seleccionados' (2 selected).

Y por último solo debemos revisar que la aplicación este correcta y se procede a crear el Kubernetes.

The screenshot shows the 'Revisar y crear' (Review and create) tab in the 'Crear un clúster de Kubernetes' (Create a Kubernetes cluster) wizard. A green banner at the top indicates 'Validación superada' (Validation passed). The page includes a search bar at the top and a navigation menu. The main content area has a tabbed interface with 'Revisar y crear' selected. Below the tabs, there is a summary of the configuration details for the cluster, organized into sections: 'Datos básicos' (Basic data), 'Grupos de nodos' (Node groups), 'Acceso' (Access), 'Redes' (Networks), and 'Integraciones' (Integrations). At the bottom, there are buttons for '< Anterior' (Previous), 'Siguiente >' (Next), and 'Crear' (Create), along with a link to 'Descargar una plantilla para la automatización' (Download a template for automation).

Datos básicos	
Suscripción	Azure subscription 1
Grupo de recursos	rg-dummy-bhd
Región	East US 2
Nombre del clúster de Kubernetes	kube-bhd
Versión de Kubernetes	1.26.6
Actualización automática	Revisión

Grupos de nodos	
Grupos de nodos	1
Habilitar nodos virtuales	Deshabilitado

Acceso	
Identidad del recurso	Identidad administrada asignada por el sistema
Cuentas locales	Habilitado
Autenticación y autorización	Cuentas locales con RBAC de Kubernetes
Tipo de cifrado	(Predeterminado) Cifrado en reposo con una clave administrada por la plataforma

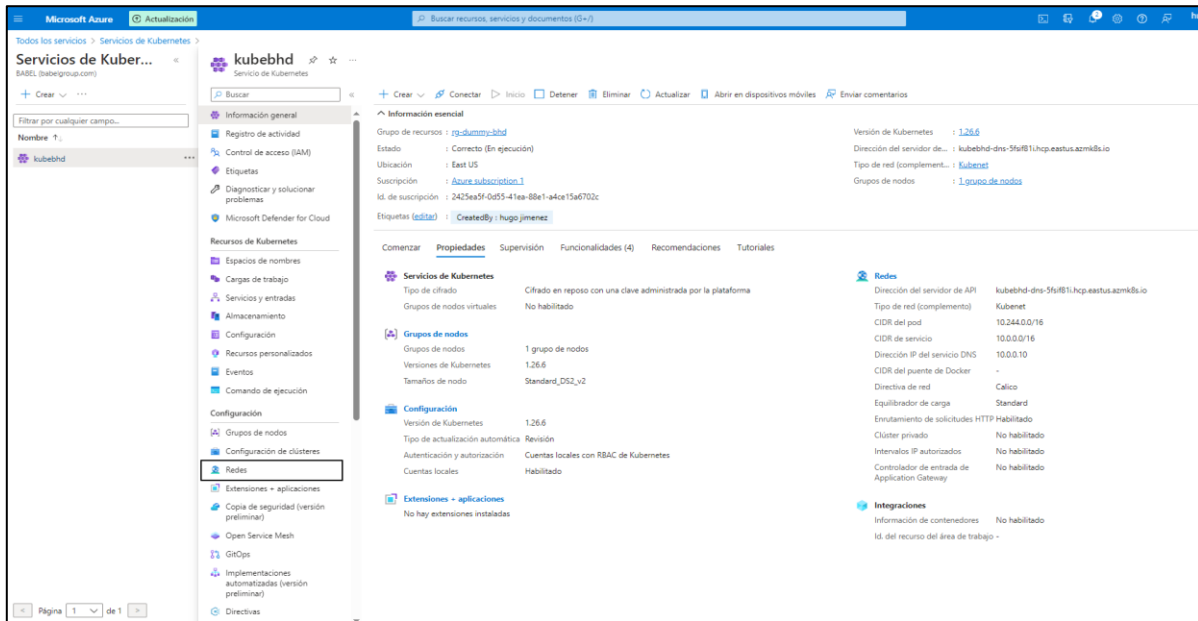
Redes	
Clúster privado	Deshabilitado
Intervalos IP autorizados	Deshabilitado
Conexión de red	Kubenet
Prefijo de nombre DNS	kube-bhd-dns
Directiva de red	Calico
Equilibrador de carga	Standard

Integraciones	
Registro de contenedor	bhdcontskube

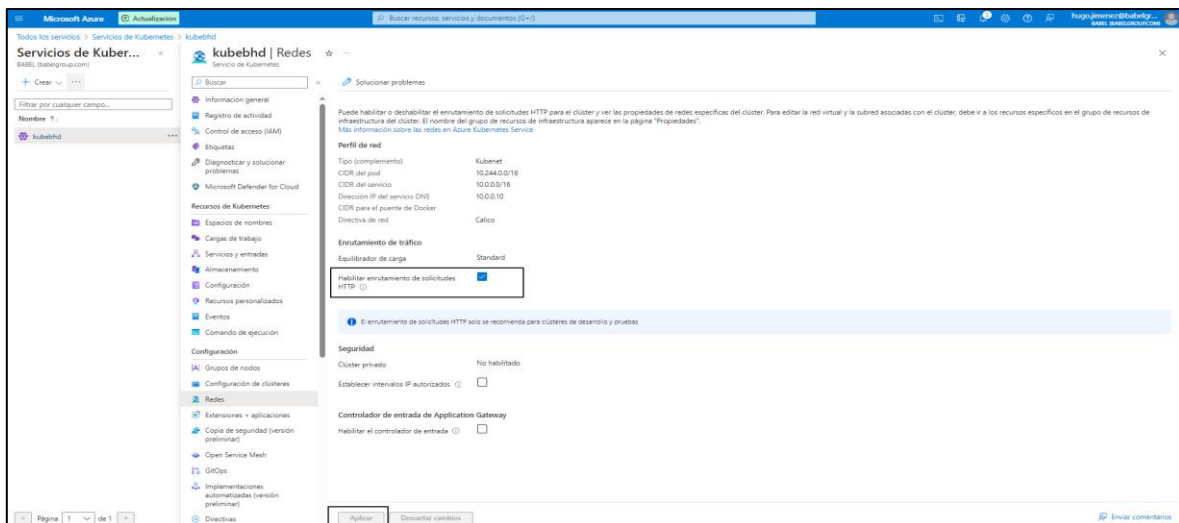


Una vez creado el Kubernetes debemos asignar la opción que permita el balanceo, esto es muy importante para poder levantar la aplicación.

En este caso procedemos a dirigirnos a redes



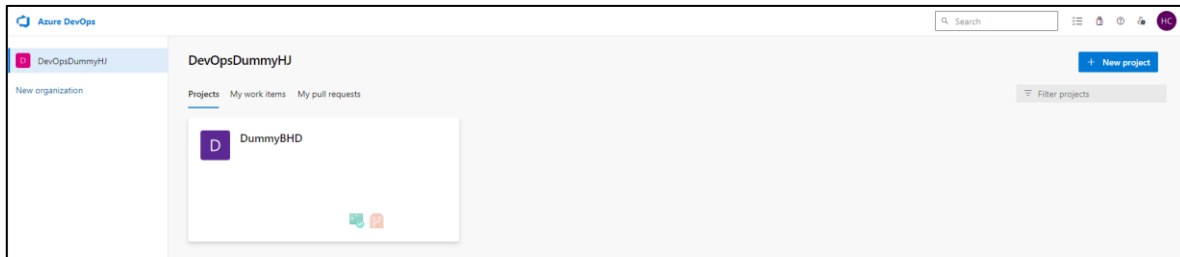
Una vez en la sección de redes, debemos habilitar la opción Habilitar enrutamiento de solicitudes HTTP y aplicar el cambio, y con esto queda listo nuestro Kubernetes para los despliegues.



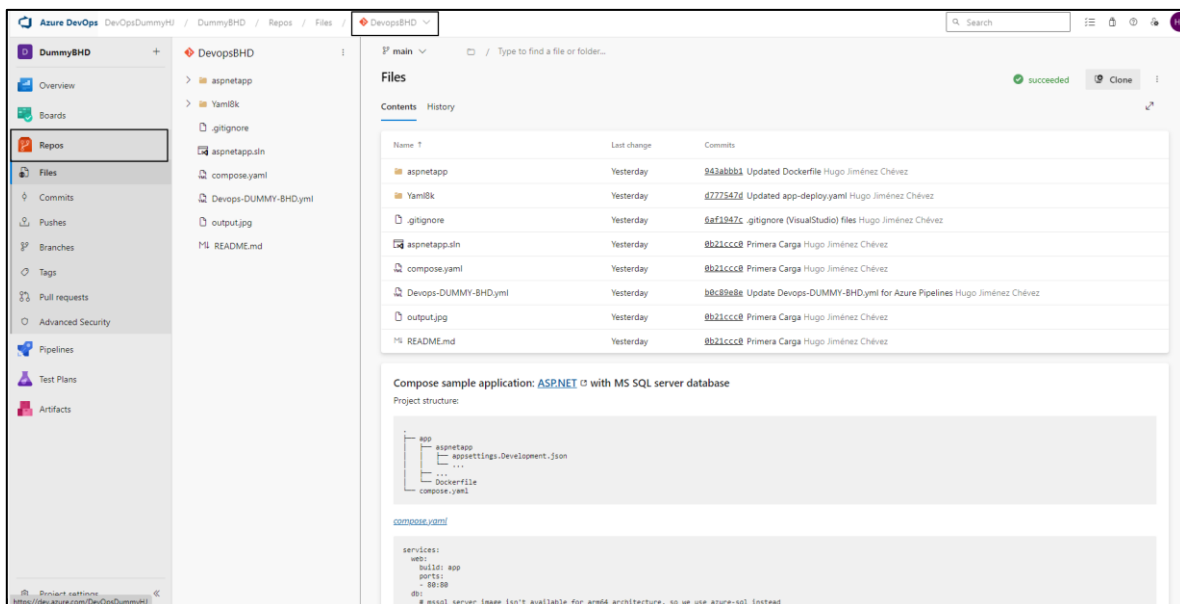


3. Azure DevOps

Primeramente, debemos crear un team Project donde realizaremos todas las configuraciones de los despliegues.

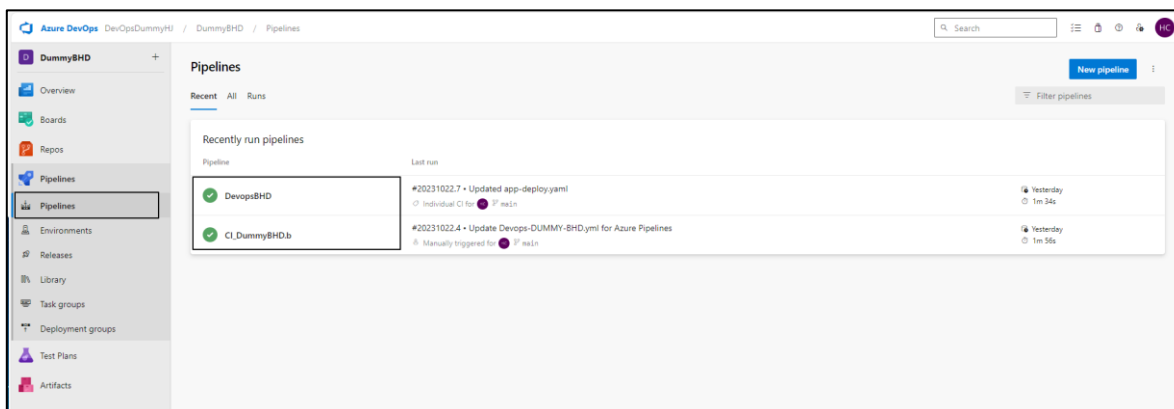


Con anterioridad se debió versionar el código fuente proporcionado para la prueba, ya que este viene con las configuraciones necesarias de Docker.



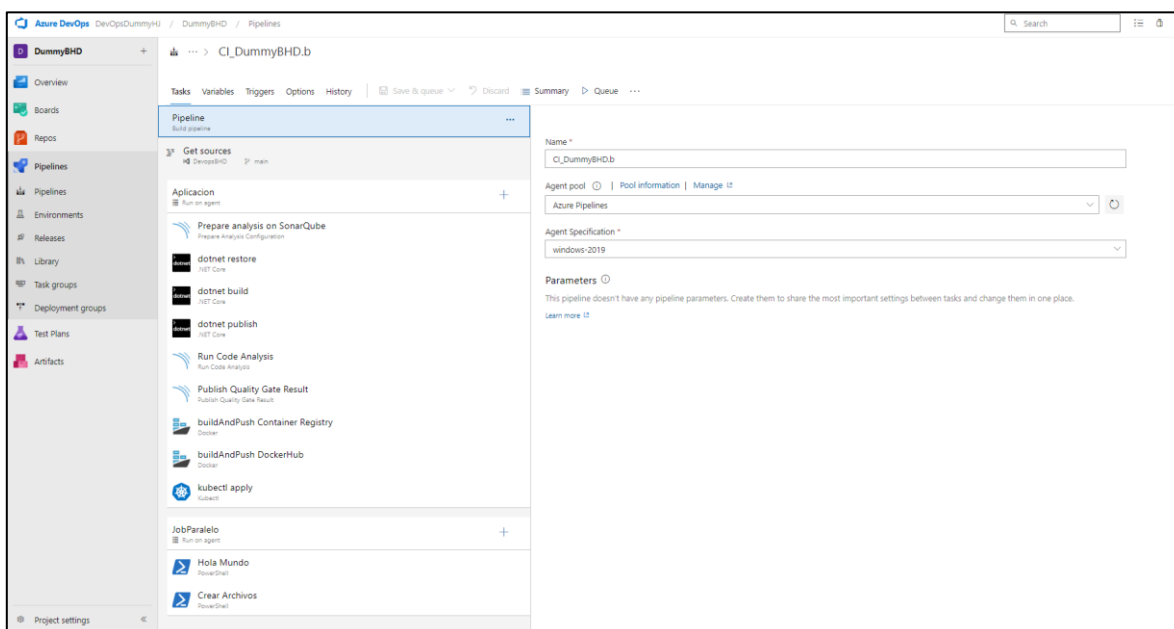


Posteriormente, se procede a configurar los pipelines.

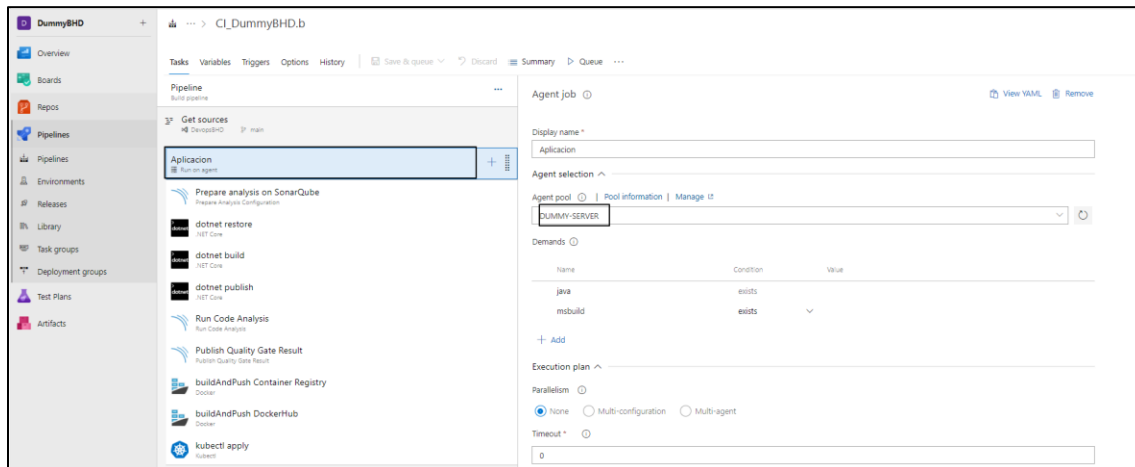


3.1. Pipeline

3.1.1. Pipeline con tareas de azure

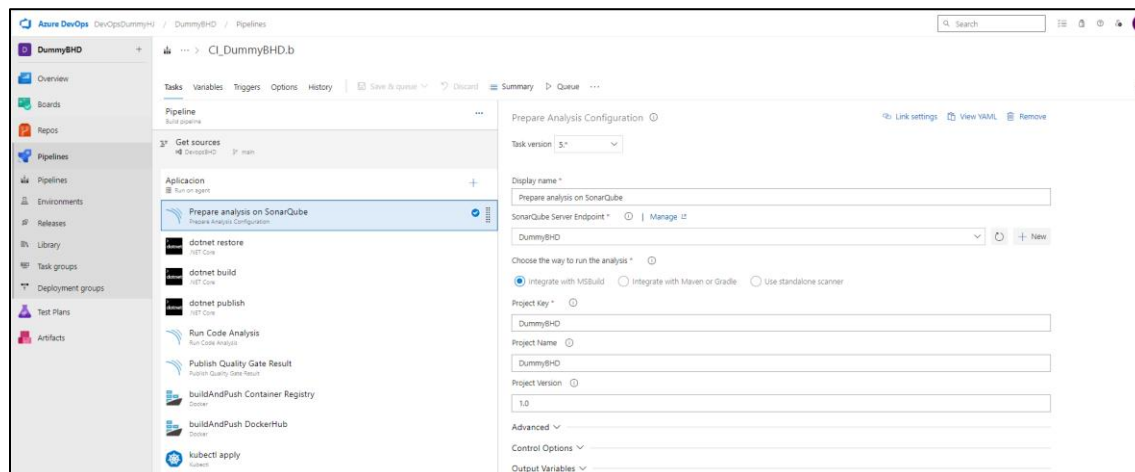


Como primer paso se asignó el Azure Agent Pool SelfHosted, ya que el agente posee todas los complementos para compilar las tareas.



SonarQube

Se optó por comenzar utilizando la tarea de SonarQube, la cual se configuró inicialmente para establecer la comunicación con SonarQube, donde se publicarán los resultados del análisis.



Para realizar la comunicación, se tuvo que agregar la siguiente configuración de la cual nos solicitaba.

Server Url: Se debe agregar la url que levanta la página de Sonarqube



Token: En Sonarqube podemos generar el token que nos solicitan

Services Connection name: Nombre de la conexión que le queremos asignar.

Grant access: Se debe marcar la opción para tener permisos desde azure.

New service connection [X]

Server Url
[Text Field]
Url for the SonarQube Server to connect to.

Authentication

Token
[Text Field]

Details

Service connection name
[Text Field]

Description (optional)
[Text Field]

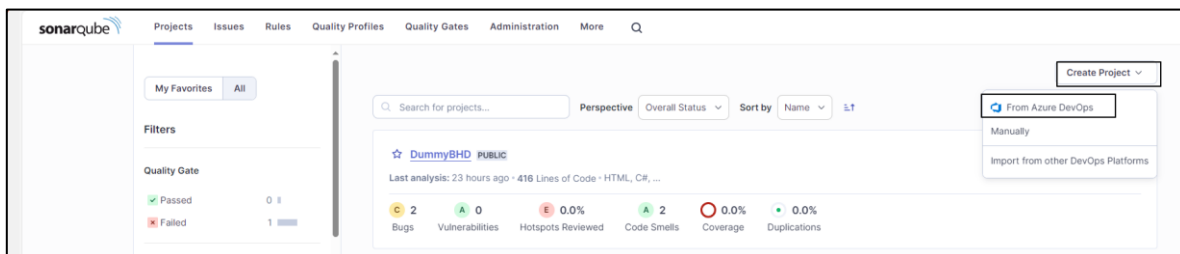
Security
☐ Grant access permission to all pipelines

[Learn more](#)
[Troubleshoot](#)

[Save]

Para generar el token, debemos realizar lo siguiente:

Debemos crear un nuevo proyecto en Sonarqube, en este caso que sea para Azure DevOps.





Si ya lo tenemos creado, solo le damos en "import" para crear el nuevo proyecto.

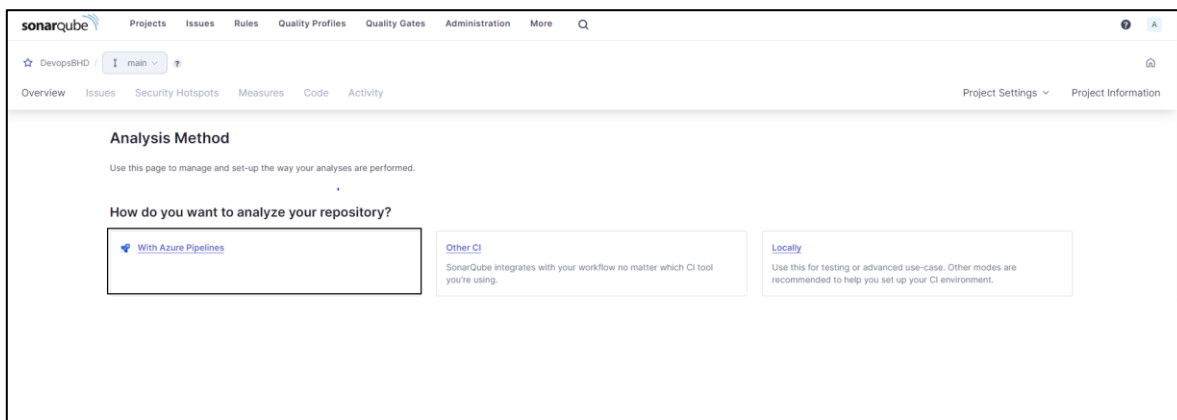
This screenshot shows the 'Azure project onboarding' page in SonarQube. The page has a search bar at the top with the text 'Search for projects and repositories'. Below it, there is a dropdown menu currently showing 'DummyBHD'. Underneath the dropdown, a list of projects is displayed, with 'DevopsBHD' being the only visible entry. To the right of this list is an 'Import' button. At the bottom of the list, it indicates '1 of 1 shown'.

En esta sección elegiremos la opción que decidamos y le damos en crear proyecto.

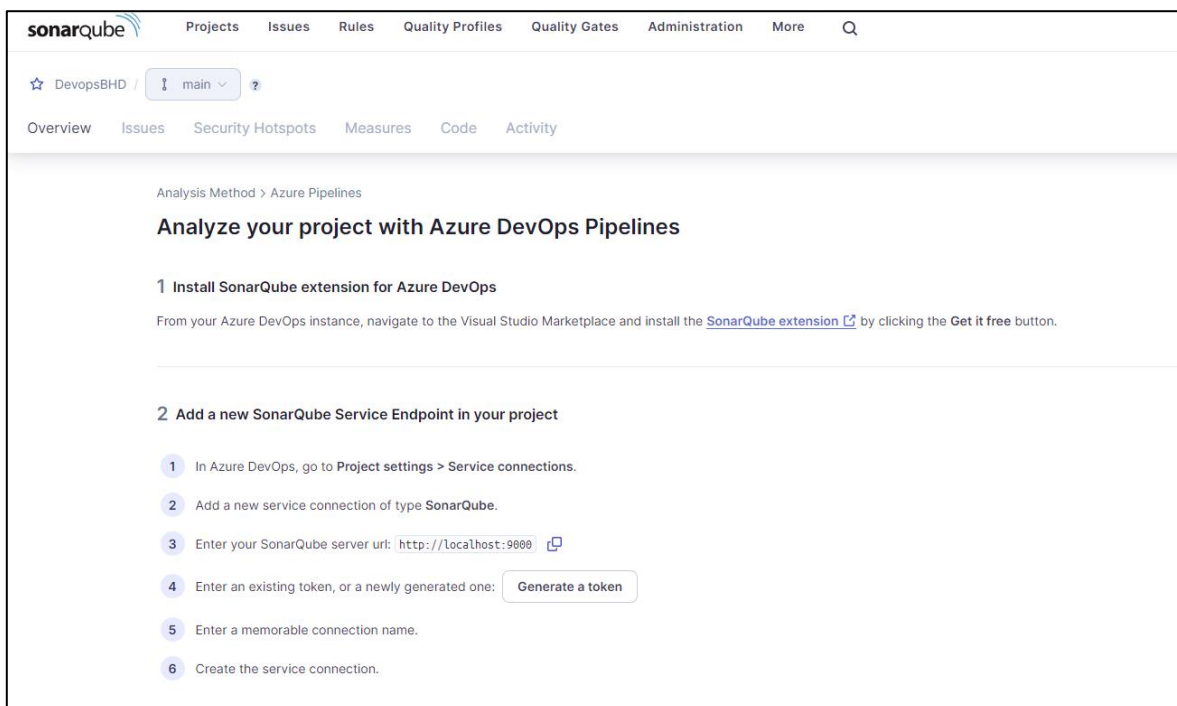
This screenshot shows the 'Set up project for Clean as You Code' page in SonarQube. The page explains the 'Clean as You Code' methodology and asks the user to 'Choose the baseline for new code for this project'. There are three main radio button options: 'Use the global setting', 'Define a specific setting for this project' (which is selected), and 'Reference branch'. Under 'Define a specific setting for this project', there are three sub-options: 'Previous version', 'Number of days', and 'Reference branch'. Each sub-option has a brief description and a recommendation. At the bottom of the page, there is a 'Create project' button.



Posteriormente, en la siguiente pestaña seleccionamos azure pipelines para hacer la comunicación.



Una vez realizadas los pasos anteriores, nos saldrá la información necesaria para crear el token.





Para generar el token, agregamos la siguiente información.

Token Name: Nombre del token.

Expires In: Cuanto tiempo expirara el token.

Y le damos en generar y copiamos el token y una vez realizado estos pasos le damos en continuar.

Generate a global token

The global token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

Token name

Expires in

Analyze "DevopsBHD"

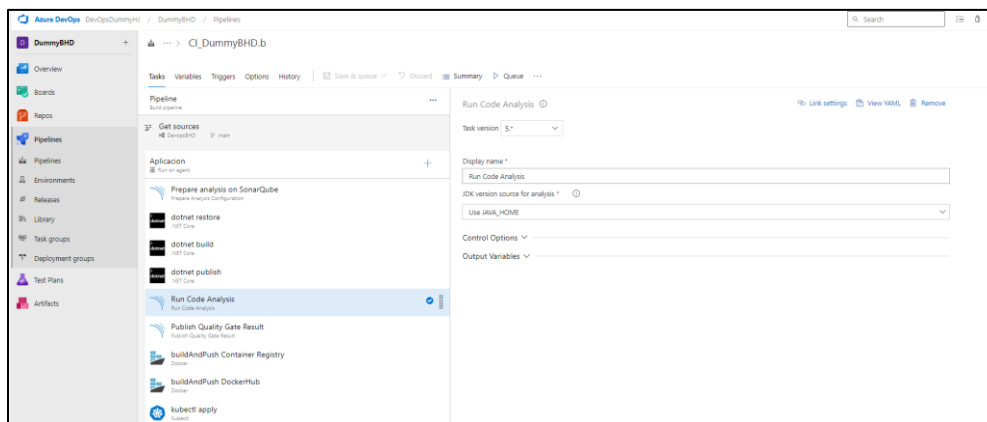
30 days ▾

Generate

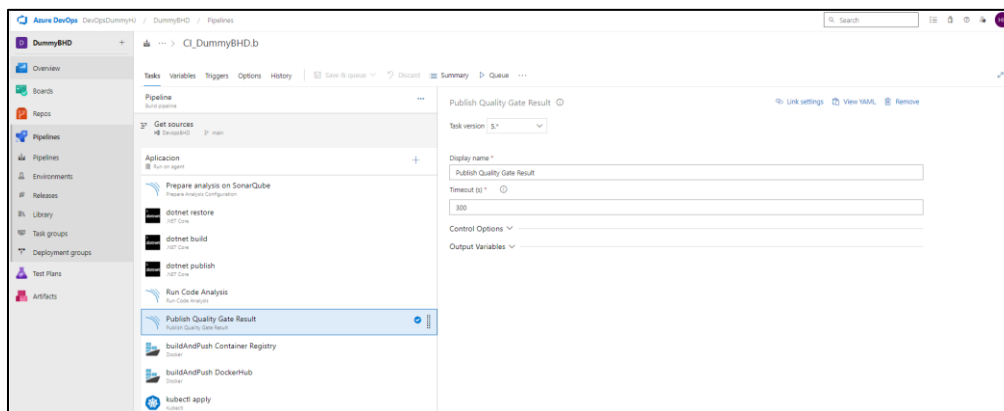
Continue



Luego, para terminar de configurar Sonarqube, se agregan las siguientes tareas: “Run Code Analysis”, que nos permitirá escanear el código compilado.

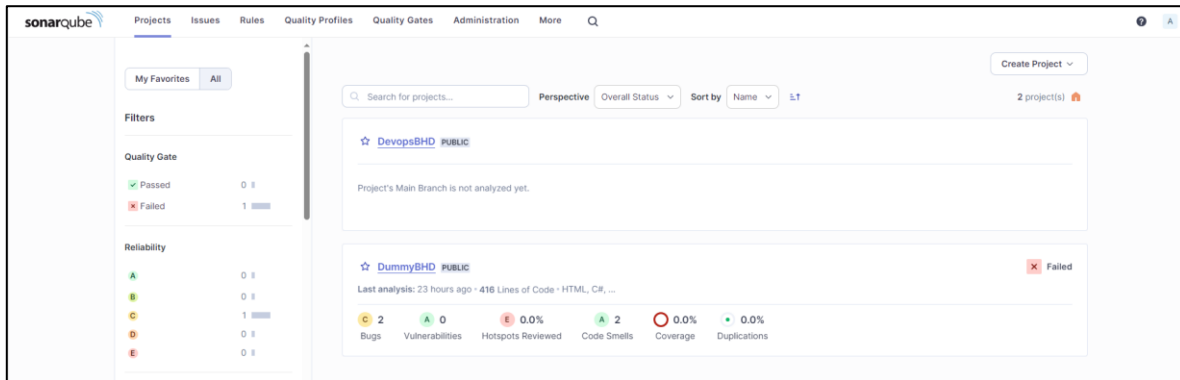


Y además, utilizaremos la tarea de “Publish Quality gate Result” que es la que se encargara de publicar los resultados en Sonarqube.



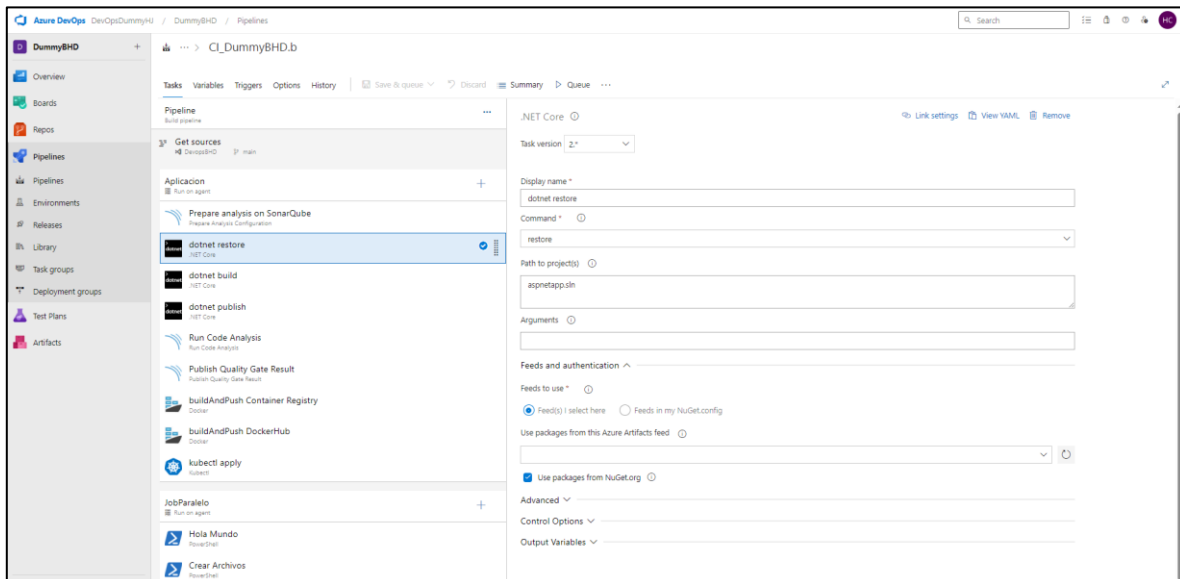


Una vez terminado las tareas de Sonarqube, al ejecutar deberíamos poder observar lo siguiente en la herramienta de sonar.



Compilador de solución.

Para compilar la solución se utilizó las tareas de Netcore (Ya que el proyecto utilizado es .Net 5)



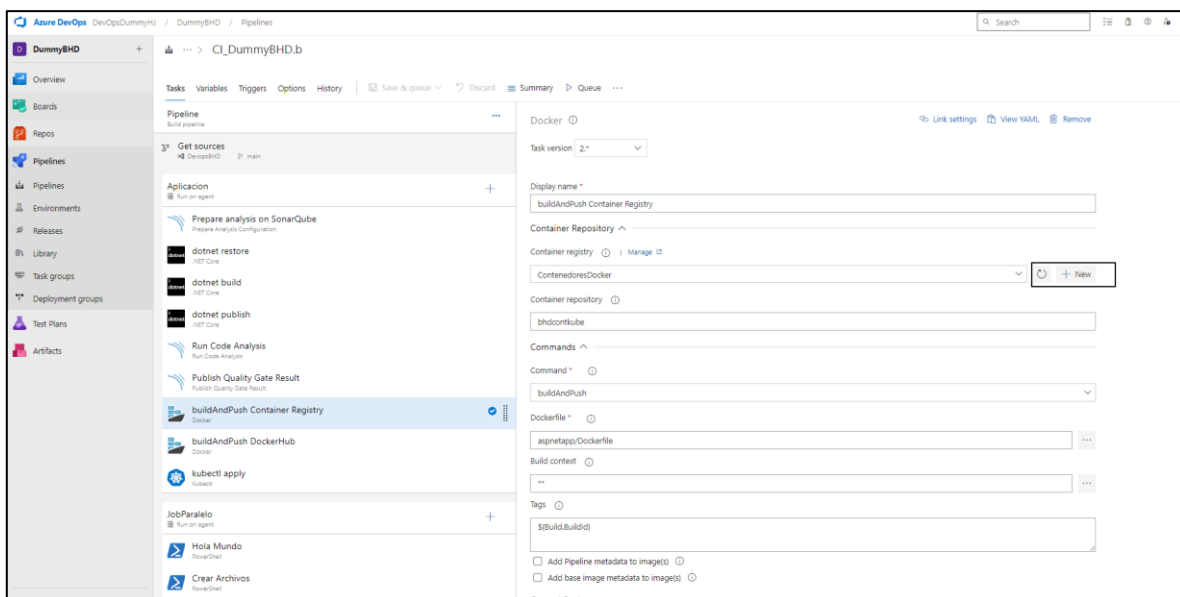


Las tareas son “dotnet restore” que es la que permite restaurar los paquetes NuGet, “dotnet build” que es la tarea que se encarga de compilar la solución y “dotnet publish” que se encarga del empaquetado.

Docker

Container Registry.

Una vez que hayamos compilado la solución, procederemos con la publicación de la imagen Docker, por lo que usaremos la siguiente tarea para convertir la aplicación en imagen y publicar en los contenedores.



En la tarea de Docker nos pide la siguiente información:

Container Registry: Es la conexión que debemos configurar para la comunicación entre azure DevOps y el contenedor.

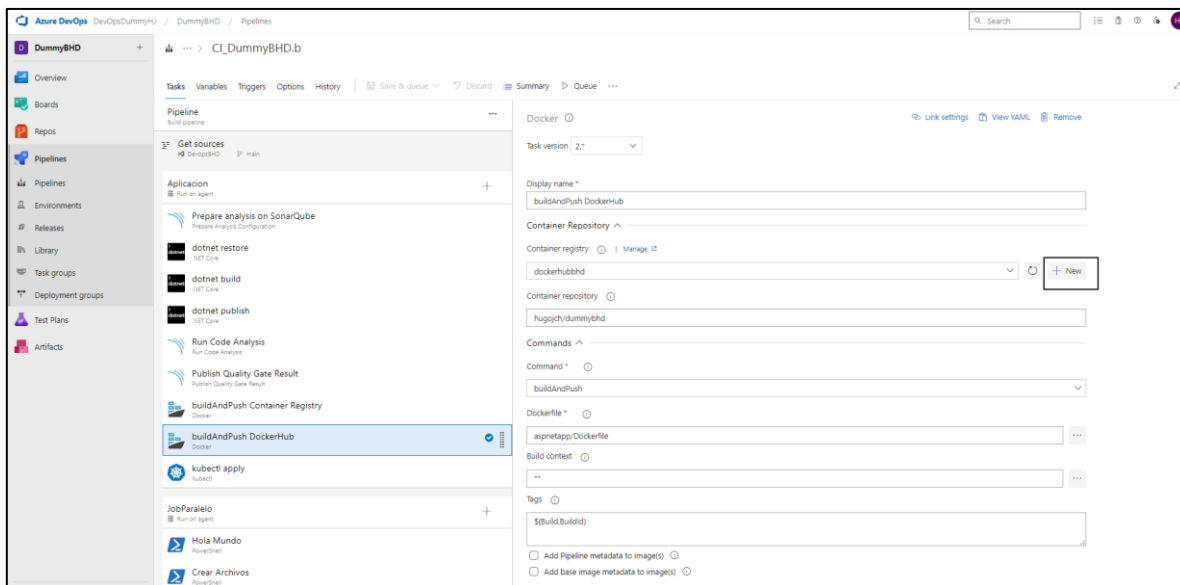
Container repository: Es el nombre que le asignamos al contenedor en Azure.

Commands: La acción a ejecutar.



Docker Hub

Una vez que hayamos compilado la solución, procederemos con la publicación de la imagen Docker, por lo que usaremos la siguiente tarea para convertir la aplicación en imagen y publicar en los contenedores. En este caso publicaremos en Docker Hub.



En la tarea de Docker nos pide la siguiente información:

Container Registry: Es la conexión que debemos configurar para la comunicación entre azure DevOps y Docker hub.

Container repository: Es el nombre que le asignamos al contenedor en Docker hub.

Commands: La acción a ejecutar.

Dockerfile: El archivo Docker que se encarga de transformar la solución en una imagen.

Para la conexión debemos rellenar con los siguiente datos



New service connection

Registry type

☒ Docker Hub ☐ Others ☐ Azure Container Registry

Docker Registry

Docker ID

Docker Password

Email (optional)

Verify

Details

Service connection name

Description (optional)

Security

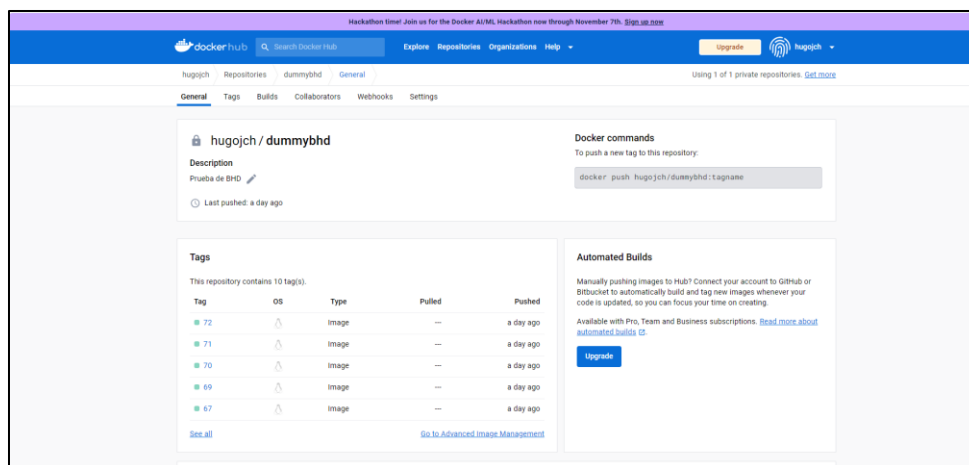
☐ Grant access permission to all pipelines

Learn more

Troubleshoot

Verify and save

Una vez que hayamos configurado la conexión y al ejecutar el pipeline, en nuestro contenedor deberían aparecer las imágenes correspondientes.



Si lo que necesitamos es validar que la imagen haya quedado funcional, debemos hacer lo siguiente.

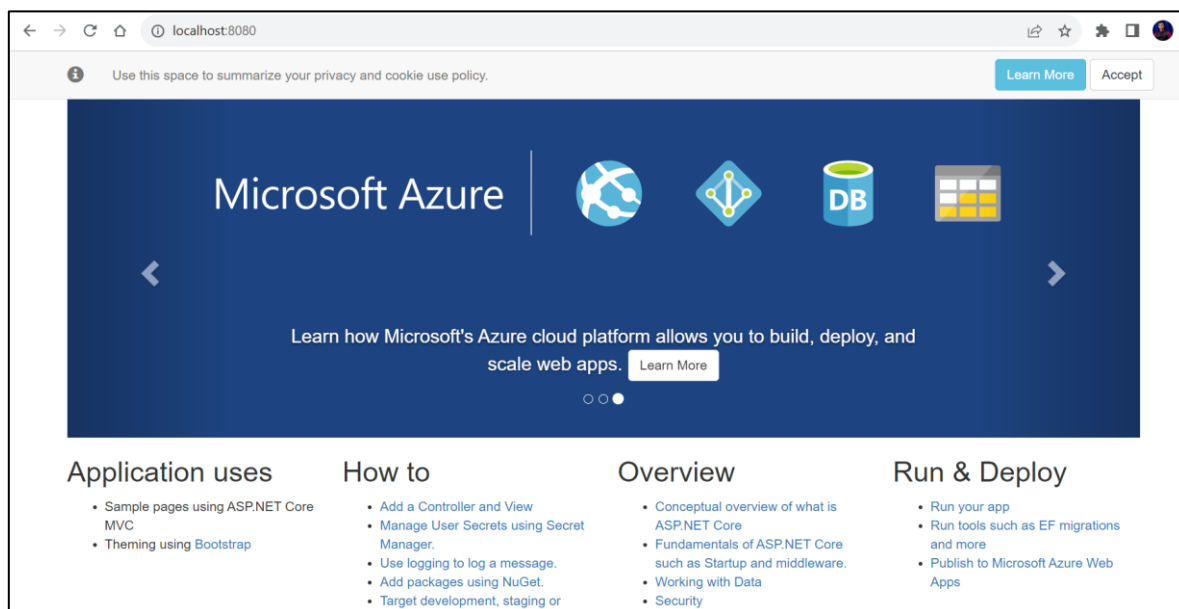
Debemos mantener en ejecución la aplicación de escritorio de Docker, luego abrimos un cmd y ejecutamos los siguientes comandos.



```
docker push hugojch/dummybhd:tagname
```

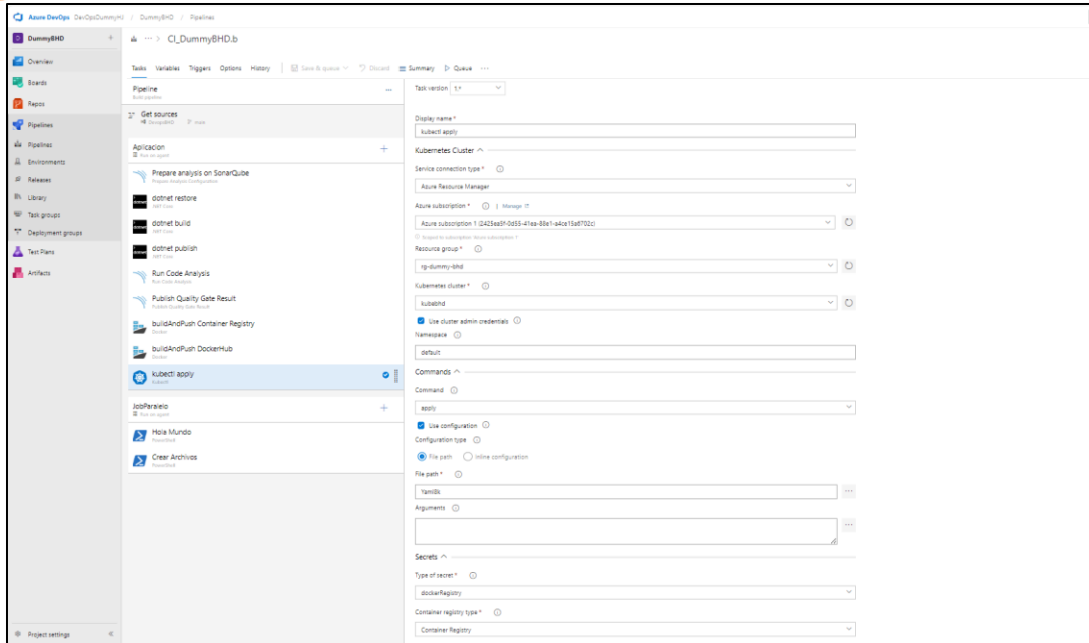
```
docker run -d -p 8080:80 --name appnet hugojch/dummybhd:74.
```

Con esto lo que hacemos es levantar el Docker por localhost y el puerto asignado.



Kubernetes

Una vez que hayamos generado la imagen y subida al container Registry, procederemos con la publicación de la imagen Docker a Kubernetes por usaremos la siguiente tarea (kubectl).



En la tarea de Kubernetes nos pide la siguiente información:

Services Connction type: Es la conexión que necesitaremos crear para la comunicación entre Azure Devops y Kubernetes.

Azure Subscription: Es la suscripción donde está alojado el Kubernetes.

Resource group: Grupo de recursos asignado del Kubernetes.

Kubernetes cluster: Kubernetes creado para la publicación de las imágenes.

Use cluster admin credentials: Se debe marcar esta opción.

Namespace: En este caso usaremos el default.

Command: Acción que deseamos que se realice.

Use configuration: Se debe marcar esta opción para utilizar el yaml que contiene ciertos valores a utilizar.

File path: Carpeta donde está alojado el yaml.



Secret: Configuración del contenedor.

Type of Secret: En este caso usaremos Container Registry.

Container Registry name: Nombre del contenedor donde se encuentran las imágenes

Force update secret: Se utiliza para forzar la actualización de las imágenes.

Con respecto al yaml utilizado, esto fue lo que se configuro con el fin de crear ciertas características para el funcionamiento del Kubernetes.

A screenshot of the Azure DevOps web interface. The left sidebar shows the project structure with 'DummyBHD' selected. The main pane displays the 'app-deploy.yaml' file. The file content is a Kubernetes manifest for a Deployment and a Service. The Deployment section (lines 1-31) defines a deployment named 'bhdcontkub' with 1 replica, using the 'bhdcontkub.azurecr.io/bhdcontkub:62' image. The Service section (lines 32-44) defines a 'LoadBalancer' service named 'bhdcontkub' on port 80, selecting the 'bhdcontkub' pods.

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: bhdcontkub
5 spec:
6   replicas: 1
7   selector:
8     matchLabels:
9       app: bhdcontkub
10  strategy:
11    rollingUpdate:
12      maxSurge: 1
13      maxUnavailable: 1
14  minReadySeconds: 5
15  template:
16    metadata:
17      labels:
18        app: bhdcontkub
19    spec:
20      nodeSelector:
21        "kubernetes.io/os": linux
22      containers:
23        - name: bhdcontkub
24          image: bhdcontkub.azurecr.io/bhdcontkub:62
25          ports:
26            - containerPort: 80
27          resources:
28            requests:
29              cpu: 250m
30            limits:
31              cpu: 500m
32 ---
33 apiVersion: v1
34 kind: Service
35 metadata:
36   name: bhdcontkub
37 spec:
38   type: LoadBalancer
39   ports:
40     - port: 80
41   selector:
42     app: bhdcontkub
43
44
```

Además, este yaml nos crea un load balancer que nos permitirá sacar la url pública.

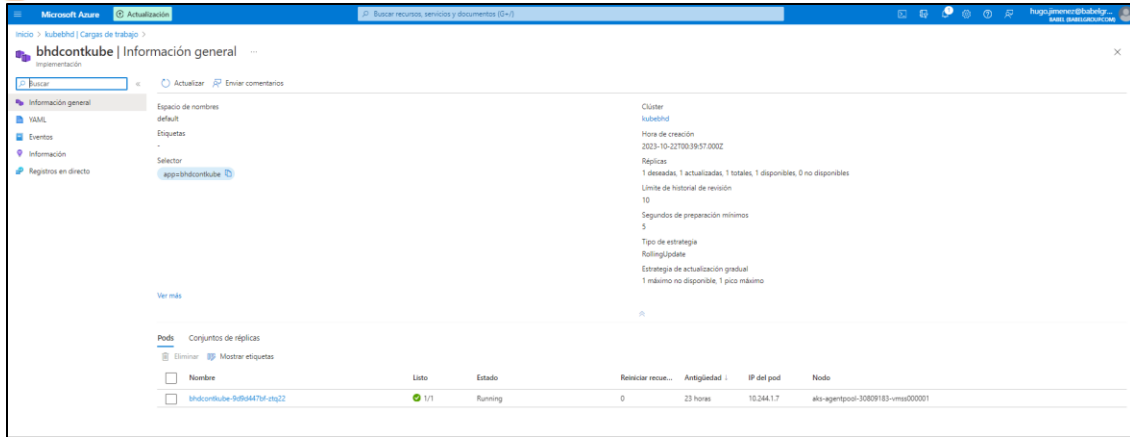


The screenshot shows the Azure DevOps interface. On the left, a sidebar contains navigation links: Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The main area displays 'Jobs in run #20231023.2' for 'C:\DummyBHD.b'. A list of jobs is shown, including 'Initialize job', 'Checkout DevOpsBHD@main to s', 'Prepare analysis on SonarQube', 'dotnet restore', 'dotnet build', 'dotnet publish', 'Run Code Analysis', 'Publish Quality Gate Result', 'buildAndPush Container Reg...', 'buildAndPush DockerHub', 'kubectl apply', 'Post-job: Checkout DevOpsBHD...', 'Finalize Job', and 'JobParalelo'. The 'kubectl apply' job is highlighted. On the right, a terminal window shows the output of the 'kubectl apply' command, displaying Kubernetes manifest details for a service named 'dummybhd'.

Una vez que hayamos configurado la tarea y al ejecutar el pipeline, en nuestro Kubernetes deberían aparecer las imágenes correspondientes

The screenshot shows the Microsoft Azure portal interface. The left sidebar contains navigation links: Información general, Registro de actividad, Control de acceso (IAM), Roles, Diagnóstico y solución de problemas, Microsoft Defender for Cloud, Recursos de Kubernetes, Espacios de nombres, Cargas de trabajo, Servicios y entornos, Almacenamiento, Configuración, Recursos personalizados, Eventos, Comandos de ejecución, and Configuración. The main area displays 'Cargas de trabajo' for 'kubebhd'. A table lists the jobs, including 'kubebhd' and 'kubebhd-kube-system'. The table has columns: Nombre, Espacio de nombres, Lista, Actualizado, Disponible, and Antigüedad. The 'kubebhd' job is highlighted.

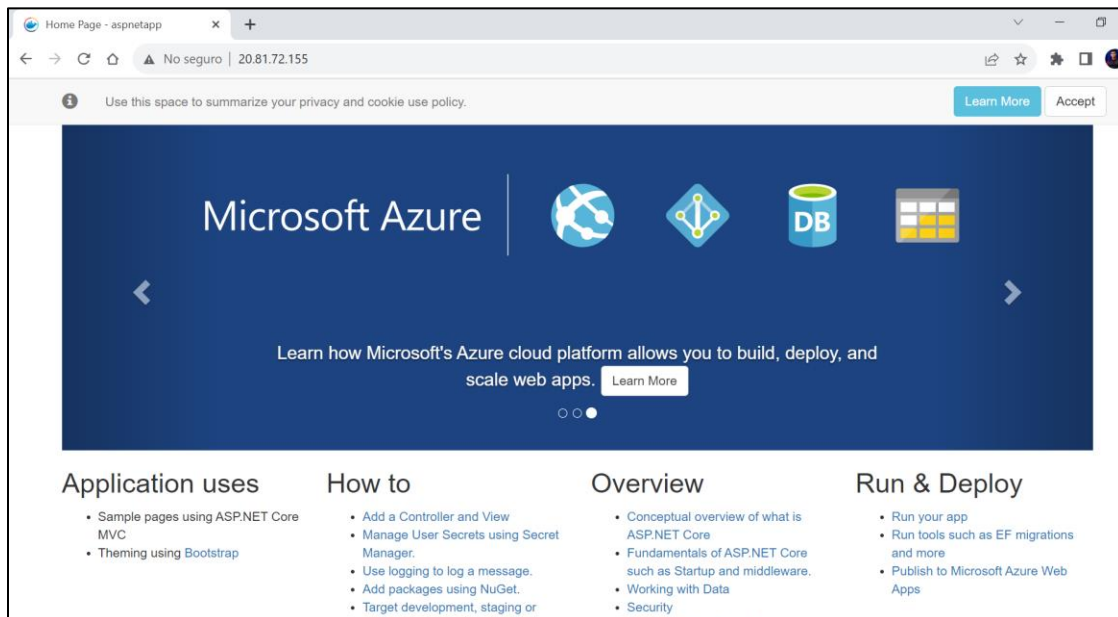
Nombre	Espacio de nombres	Lista	Actualizado	Disponible	Antigüedad
coredns	kube-system	2/2	2	2	1 día
coredns-autoscaler	kube-system	1/1	1	1	1 día
connectivity-agent	kube-system	2/2	2	2	1 día
metrics-server	kube-system	2/2	2	2	1 día
nginx-operator	nginx-operator	1/1	1	1	1 día
calico-kube-controllers	calico-system	1/1	1	1	1 día
calico-typha	calico-system	1/1	1	1	1 día
nginx-kube	default	1/1	1	1	1 día
addon-http-application-routing-external-dns	kube-system	1/1	1	1	1 día
addon-http-application-routing-nginx-ingress-controller	kube-system	1/1	1	1	1 día



Para validar que la aplicación quedo publicada y funcional, procedemos a ingresar por medio del load balancer configurado en el yaml anterior.

<http://20.81.72.155/>

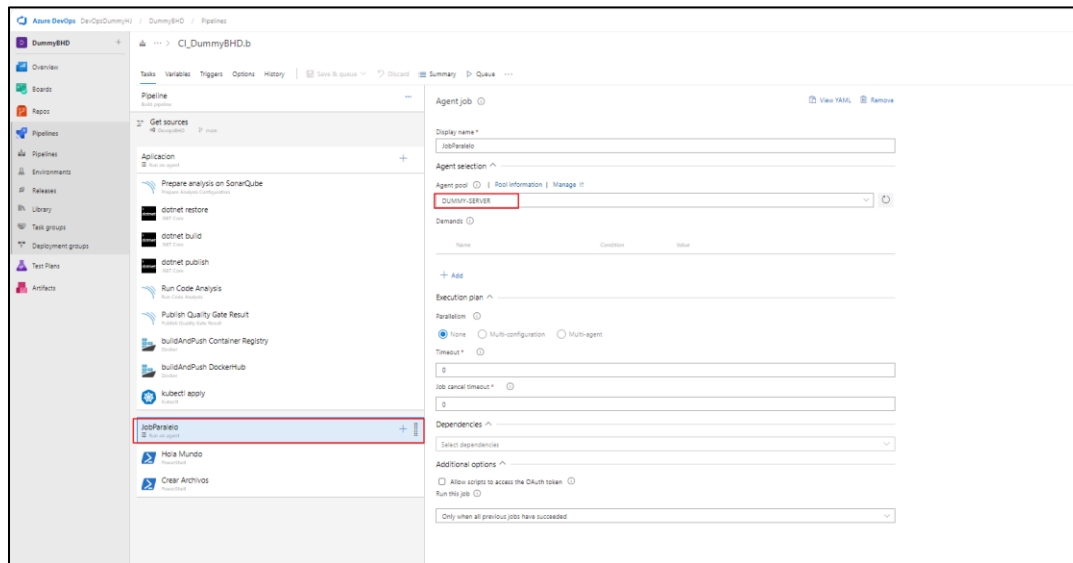
Y de esta forma podemos observar que la publicación fue exitosa.



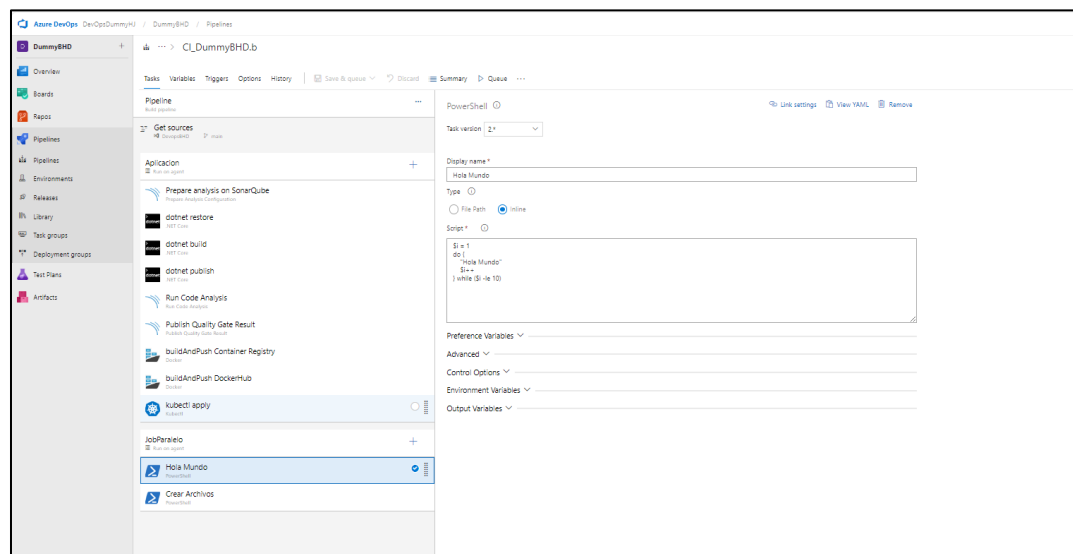


Configuración Job paralelo.

Como primer paso se asignó el Azure Agent Pool SelfHosted, ya que el agente posee todas los complementos para compilar las tareas.

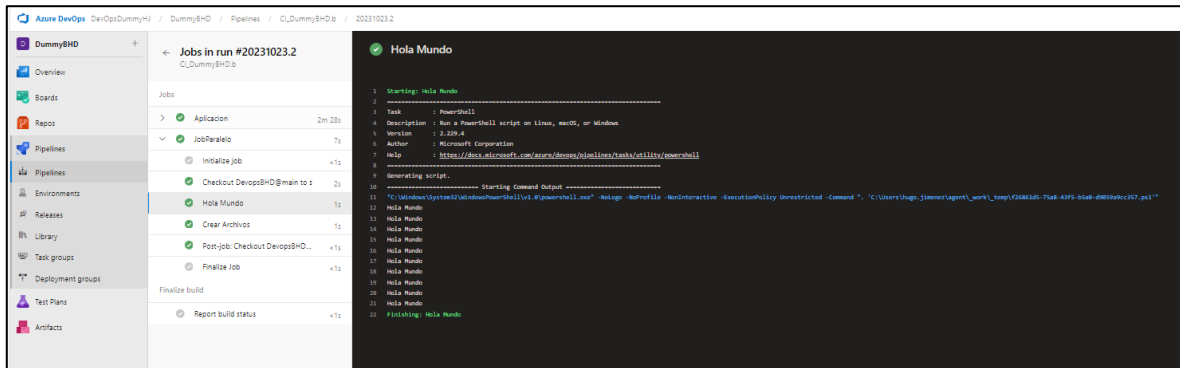


Seguidamente, se configuro un script por medio de PowerShell que imprime 10 veces el "hola mundo"

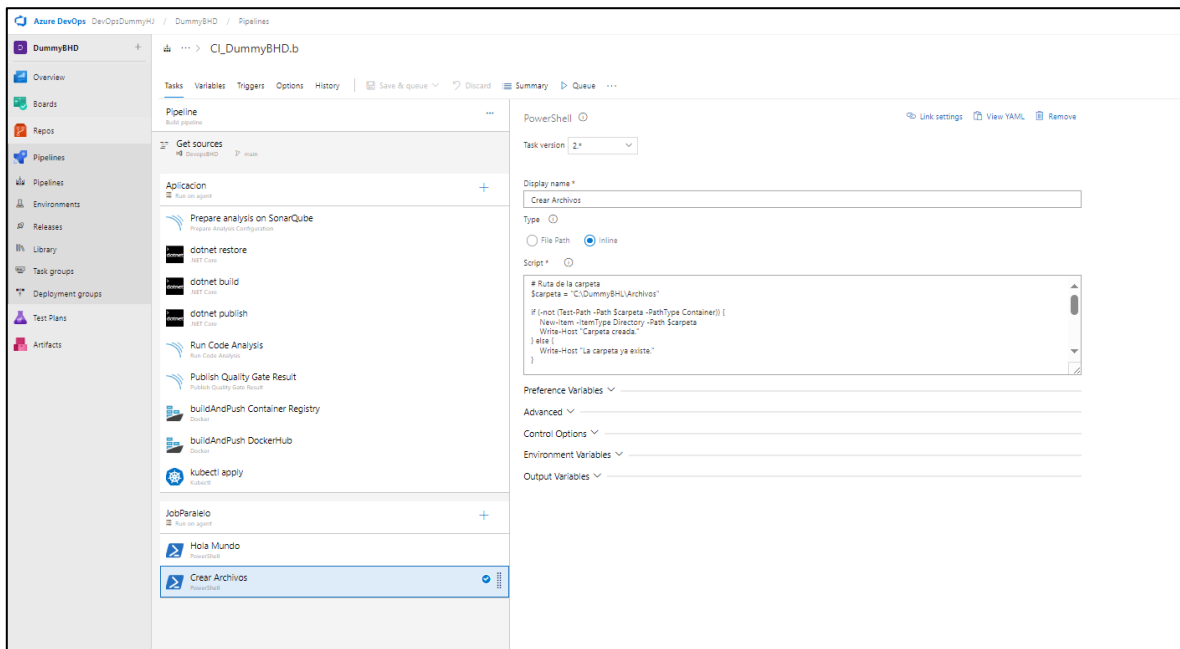




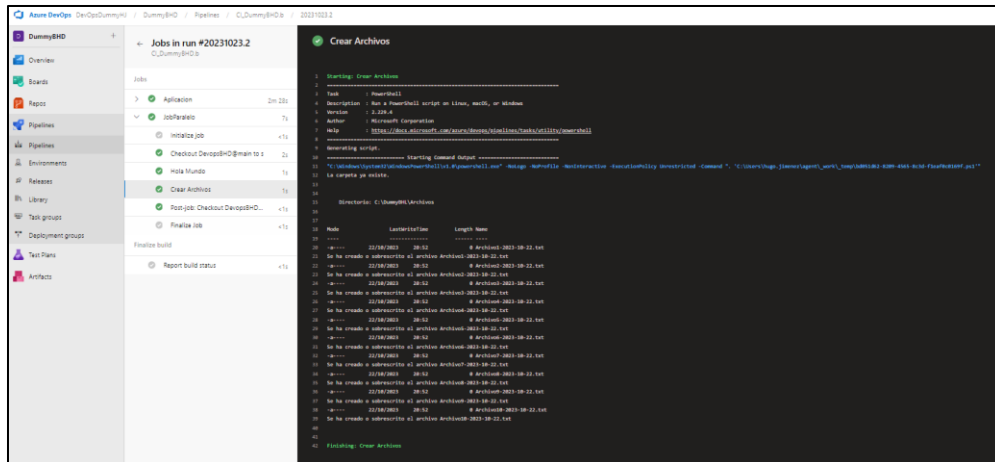
A continuación, en los logs podemos observar que efectivamente se imprimen los hola mundo.



Posteriormente, se procede a crear un script que crea 10 archivos con la fecha y los imprime



Y como podemos observar a continuación los archivos con las fechas.

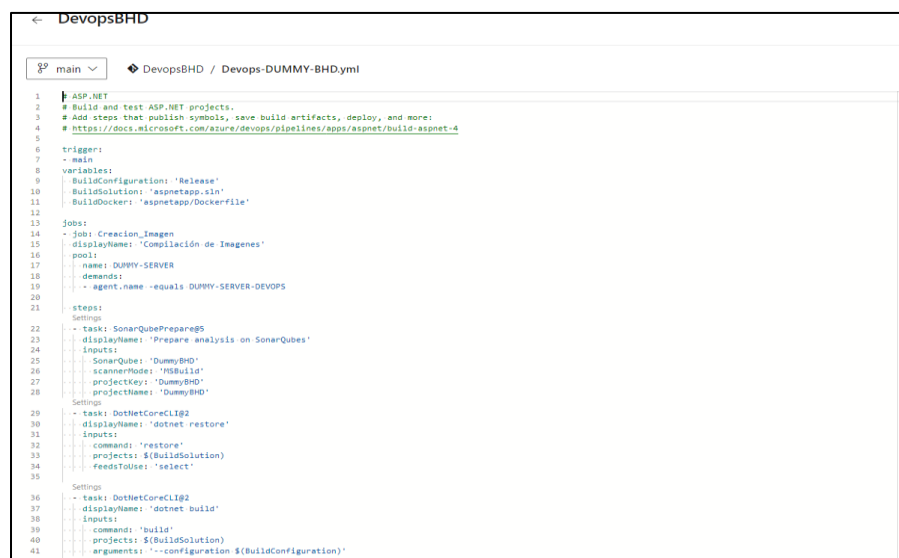


3.1.2. Pipeline yaml

Aparte de las tareas de azure, se creo otro pipeline, solo que esta vez utilizando yaml para configuración del proceso devops, las configuraciones son las mismas que lo explicado anteriormente.

Jobs 1

Se configuro las tareas en yaml para la construcción de la imagen Docker y su despliegue hacia Kubernetes.





```
← DevopsBHD

main ▾ DevopsBHD / Devops-DUMMY-BHD.yml

43 - task: Docker@2
44   displayname: 'docker publish'
45   inputs:
46     command: 'publish'
47     publishBuildProject: false
48     project: $(BuildId)
49     arguments: '-configuration $(BuildConfiguration) --output $(Build.ArtifactStagingDirectory)'
50     zipAfterPublish: false
51
52 - task: SonarQubeAnalyze@2
53   displayname: 'Run Code Analysis'
54   inputs:
55     | server: 'TARA_HOME'
56
57 - task: SonarQubePublish@2
58   displayname: 'Publish Quality Gate Result'
59   inputs:
60     | pollingTimeout: '300'
61
62 - task: Docker@2
63   displayname: 'buildAndPush Container Registry'
64   inputs:
65     | containerRegistry: 'ContainerRegistry'
66     | repository: 'bhd:latest'
67     | command: 'buildAndPush'
68     | dockerfile: '$(BuildDocker)'
69
70 - task: Docker@2
71   displayname: 'buildAndPush DockerHub'
72   inputs:
73     | containerRegistry: 'dockerhub'
74     | repository: 'bhd:latest'
75     | command: 'buildAndPush'
76     | dockerfile: '$(BuildDocker)'
77
78 - task: Kubernetes@1
79   displayname: 'kubectl apply'
80   inputs:
81     | connectionType: 'Azure Resource Manager'
82     | azureSubscriptionEndpoint: 'Azure Subscription 1 (2425e5f-b055-43ea-b8b1-akce1a5782c)'
83     | azureResourceGroup: 'rg-dummy-bhd'
84     | kubernetesCluster: 'kubeha'
85     | namespace: 'default'
86     | command: 'apply'
87     | useConfigurationFile: true
88     | configuration: 'yaml'
89     | secureType: 'secretRegistry'
90     | containerRegistryType: 'Azure Container Registry'
91     | azureSubscriptionEndpointForSecrets: 'Azure Subscription 1 (2425e5f-b055-43ea-b8b1-akce1a5782c)'
92     | azureContainerRegistry: 'bhd:latest.azurecr.io'
```

Jobs 2

En el segundo jobs del Yaml se configuro las pruebas de imprimir 10 hola mundos y la creación de los archivos con su fecha usando PowerShell.

```
← DevopsBHD

main ▾ DevopsBHD / Devops-DUMMY-BHD.yml

93 - job: Job_Paralelo
94   displayName: 'Compilación Scripts'
95   pool:
96     name: DUMMY-SERVER
97   demands:
98     - agent.name -equals DUMMY-SERVER-DEVOPS
99   steps:
100     - task: PowerShell@2
101       displayname: 'hola Mundo'
102       inputs:
103         targetType: 'inline'
104         script: |
105           $i = 1
106           do {
107             "hola Mundo"
108             $i++
109           } while ($i -le 10)
110
111     - task: PowerShell@2
112       displayname: 'Crear Archivos'
113       inputs:
114         targetType: 'inline'
115         script: |
116           # Ruta de la carpeta
117           $carpeta = "C:\DUMMY\Archivos"
118
119           if (-not (Test-Path -Path $carpeta -PathType Container)) {
120             New-Item -ItemType Directory -Path $carpeta
121             Write-Host "Carpeta creada."
122           } else {
123             Write-Host "La carpeta ya existe."
124           }
125
126           # Obtener la fecha actual en un formato específico (por ejemplo, YYYY-MM-DD).
127           $fecha = Get-Date -Format "yyyy-MM-dd"
128
129           # Ruta donde se crearán los archivos. Cambia la ruta a tu preferencia.
130           $ruta = $carpeta
131
132           # Bucle para crear 10 archivos con nombres basados en la fecha.
133           for ($i = 1; $i -le 10; $i++) {
134             $nombreArchivo = "Archivos-$fecha.txt"
135             $rutaCompleta = Join-Path -Path $ruta -ChildPath $nombreArchivo
136
137             New-Item -Path $rutaCompleta -ItemType File -Force
138             Write-Host "Se ha creado o sobrescrito el archivo $nombreArchivo"
139           }
```