

# Entorno de Azure

Esta documentación tiene como objetivo detallar el proceso de construcción del entorno base necesario para desplegar nuestra aplicación orientada a la **gestión y organización de recursos**. Aprovecharemos las ventajas que ofrece **Azure** frente a las herramientas previamente utilizadas en **Power Platform**, permitiendo una arquitectura más escalable, flexible y desacoplada.

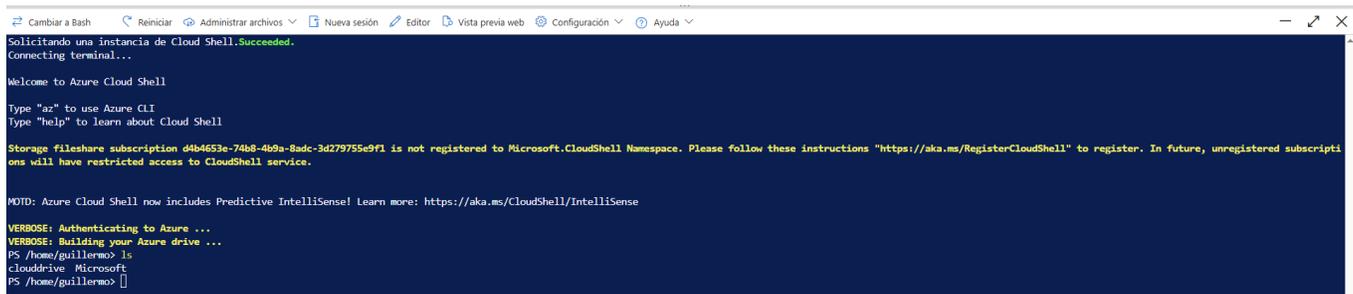
Al finalizar, habremos configurado el entorno compatible con el siguiente stack tecnológico:

1. **Azure Container Apps**
2. **Azure SQL Database**
3. **Azure Static Web Apps**
4. **Azure Power BI Embedded**
5. **Azure Active Directory (Azure AD)**
6. **Azure Container Registry**
7. **Azure VNet**

---

## 1. Despliegue de CLI

Como primer paso, accedimos a **Azure Cloud Shell**. Al iniciar, el sistema solicitó el aprovisionamiento de un espacio de almacenamiento asociado a la suscripción:



```

C Cambiar a Bash  Reiniciar  Administrar archivos  Nueva sesión  Editor  Vista previa web  Configuración  Ayuda
Solicitando una instancia de Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

Storage fileshare subscription d4b4653e-74b8-4b9a-8adc-3d279755e9f1 is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to CloudShell service.

NOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/guillermo> ls
cloudrive  Microsoft
PS /home/guillermo> ]
```

---

## 2. Registro de proveedores

Ejecutamos los siguientes comandos para registrar los proveedores que Azure requiere para los servicios que utilizaremos:

```
az provider register -n Microsoft.OperationalInsights --wait
```

```
az provider show --namespace Microsoft.CloudShell --query "registrationState"
```

Esto garantiza que los recursos como Log Analytics y Cloud Shell estén habilitados:

```
Subscription 0404653e-7408-4b9a-8a0c-5d279735e9f1 is not registered for the Microsoft
PS /home/guillermo> az provider register -n Microsoft.OperationalInsights --wait
```

```
PS /home/guillermo> az provider show --namespace Microsoft.CloudShell --query "registrationState"
"Registered"
PS /home/guillermo> |
```

---

### 3. Creando nuestro primer grupo

Creamos un **grupo de recursos** llamado `Inventory-TI` en la región `centralus`, que servirá como contenedor lógico para los servicios de Azure:

```
az group create --name Inventory-TI --location centralus
```

En caso que el resultado sea exitoso veremos:

```
PS /home/guillermo> az group create --name Inventory-TI --location centralus
{
  "id": "/subscriptions/3c2373b3-e3e3-4a82-99da-885c09671dd7/resourceGroups/Inventory-TI",
  "location": "centralus",
  "managedBy": null,
  "name": "Inventory-TI",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
PS /home/guillermo> |
```

---

### 4. Creando nuestro Log Analytics Workspace

Con el grupo de recursos listo, procedimos a crear el **entorno de Azure Container Apps**, el cual alojará nuestras aplicaciones en contenedores. Este entorno también configura

automáticamente un workspace de monitoreo con Log Analytics.

```
az monitor log-analytics workspace create --resource-group Inventory-TI --workspace-name inventory-logs --location centralus
```

Esto generará un ID único para el workspace ( `customerId` ), que necesitaremos en el siguiente paso

La salida que deberíamos ver:

```
PS /home/guillermo> az monitor log-analytics workspace create --resource-group Inventory-TI --workspace-name inventory-logs --location centralus
{
  "createdDate": "2025-08-07T20:15:57.9984222Z",
  "customerId": "9d1abf74-4651-4cd6-ba86-82ef6b5e922b",
  "etag": "\"3a022f4d-0000-0300-0000-68950aac0000\"",
  "features": {
    "enableLogAccessUsingOnlyResourcePermissions": true,
    "legacy": 0,
    "searchVersion": 1
  },
  "id": "/subscriptions/3c2373b3-e3e3-4a82-99da-885c09671dd7/resourceGroups/Inventory-TI/providers/Microsoft.OperationalInsights/workspaces/inventory-logs",
  "location": "centralus",
  "modifiedDate": "2025-08-07T20:21:00.1873266Z",
  "name": "inventory-logs",
  "provisioningState": "Succeeded",
  "publicNetworkAccessForIngestion": "Enabled",
  "publicNetworkAccessForQuery": "Enabled",
  "resourceGroup": "Inventory-TI",
  "retentionInDays": 30,
  "sku": {
    "lastSkuUpdate": "2025-08-07T20:15:57.9984222Z",
    "name": "PerGB2018"
  },
  "type": "Microsoft.OperationalInsights/workspaces",
  "workspaceCapping": {
    "dailyQuotaGb": -1.0,
    "dataIngestionStatus": "RespectQuota",
    "quotaNextResetTime": "2025-08-08T18:00:00Z"
  }
}
PS /home/guillermo>
```

## 5. Creando nuestro entorno para Azure Container Environment

Una vez creado el grupo de recursos y el workspace de monitoreo, el siguiente paso es **crear el entorno administrado para Azure Container Apps (ACA)**. Este entorno funciona como la base sobre la cual se desplegarán nuestras aplicaciones en contenedor y, además, se encargará de gestionar la infraestructura necesaria para ejecutar, escalar y monitorear los contenedores.

Para ello utilizamos el siguiente comando:

```
az containerapp env create --name inventory-env --resource-group Inventory-TI --location centralus --logs-workspace-id 9d1abf74-4651-4cd6-ba86-82ef6b5e922b
```

## 🔥 Consiguiendo el ID

El ID de nuestro WorkSpace lo vamos a encontrar directamente en el JSON que nos retorna la terminal, por lo que podemos reemplazar el de este ejemplo con el ID específico que se nos genere

Este entorno será el punto de integración entre nuestras apps desplegadas en ACA y otros servicios como **Log Analytics**, **Redes privadas (VNet)** o incluso **Azure SQL Database**, permitiendo una arquitectura desacoplada pero segura.

---

## 6. Creando nuestro Azure Container Register (ACR)

El siguiente paso en la preparación de nuestra infraestructura es **crear un Azure Container Registry (ACR)**. Este servicio nos permitirá almacenar y administrar nuestras imágenes de contenedor de forma segura y privada dentro de Azure.

El ACR actuará como repositorio donde publicaremos las imágenes Docker que luego serán desplegadas en nuestra **Azure Container App**.

Usamos el siguiente comando para crearlo:

```
az acr create --name inventorytiapp --resource-group Inventory-TI --sku Premium --location centralus --admin-enabled true
```

```
PS /home/guillermo> az acr create --name inventorytiapp --resource-group Inventory-TI --sku Basic --location centralus --admin-enabled true
{
  "adminUserEnabled": true,
  "anonymousPullEnabled": false,
  "autoGeneratedDomainNameLabelScope": "Unsecure",
  "creationDate": "2025-08-07T20:36:35.113682+00:00",
  "dataEndpointEnabled": false,
  "dataEndpointHostNames": [],
  "encryption": {
    "keyVaultProperties": null,
    "status": "disabled"
  },
  "id": "/subscriptions/3c2373b3-e3e3-4a82-99da-885c09671dd7/resourceGroups/Inventory-TI/providers/Microsoft.ContainerRegistry/registries/inventorytiapp",
  "identity": null,
  "location": "centralus",
  "loginServer": "inventorytiapp.azurecr.io",
  "metadataSearch": "Disabled",
  "name": "inventorytiapp",
  "networkRuleBypassOptions": "AzureServices",
  "networkRuleSet": null,
  "policies": {
    "azureAdAuthenticationAsArmPolicy": {
      "status": "enabled"
    },
    "exportPolicy": {
      "status": "enabled"
    },
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2025-08-07T20:36:48.460497+00:00",
      "status": "disabled"
    }
  }
}
```

Ya que estaremos usando VNets Más adelante es importante que lo dejemos con un valor de premium para hacer todas las configuraciones necesarias.

## 7. Usando nuestro contenedor ACR

Si ya hemos hecho las configuraciones adecuadas veremos que nuestra implementación se completó de manera exitosa:

The screenshot shows the Azure portal interface for a Microsoft.ContainerRegistry resource. The page title is "Microsoft.ContainerRegistry | Información general". The main content area displays a green checkmark and the message "Se completó la implementación". Below this, the implementation details are shown, including the name "Microsoft.ContainerRegistry", subscription "Azure subscription 1", and resource group "Inventory-TI". The start time is "5/8/2025, 11:03:39 a.m." and the correlation ID is "06c93d7b-bfde-43bc-bf05-15ba7c9c29b1".

| Recurso        | Tipo               | Estado | Detalles de la operación |
|----------------|--------------------|--------|--------------------------|
| InventoryTIApp | Container registry | OK     | Detalles de la operación |

Below the table, there is a section for "Pasos siguientes" with a button labeled "Ir al recurso". At the bottom, there is a section for "Enviar comentarios" with a link to "Cuéntenos su experiencia con la implementación".

Por lo que ahora basta con subir nuestra imagen a nuestro contenedor para que ella la almacene, para eso estaremos haciendo uso de la CLI de Azure y en caso de no tenerla deberemos instalar esta misma en nuestro sistema.

Abriremos el CMD como administrador y ejecutaremos el comando:

```
winget install --exact --id Microsoft.AzureCLI --version 2.67.0
```

Este comando está recuperado de la documentación oficial de Microsoft:

<https://learn.microsoft.com/es-es/cli/azure/install-azure-cli-windows?view=azure-cli-latest&pivots=winget>

Una vez que usemos este comando nos descargará la CLI y podremos usar los prefijos AZ para nuestros comandos:



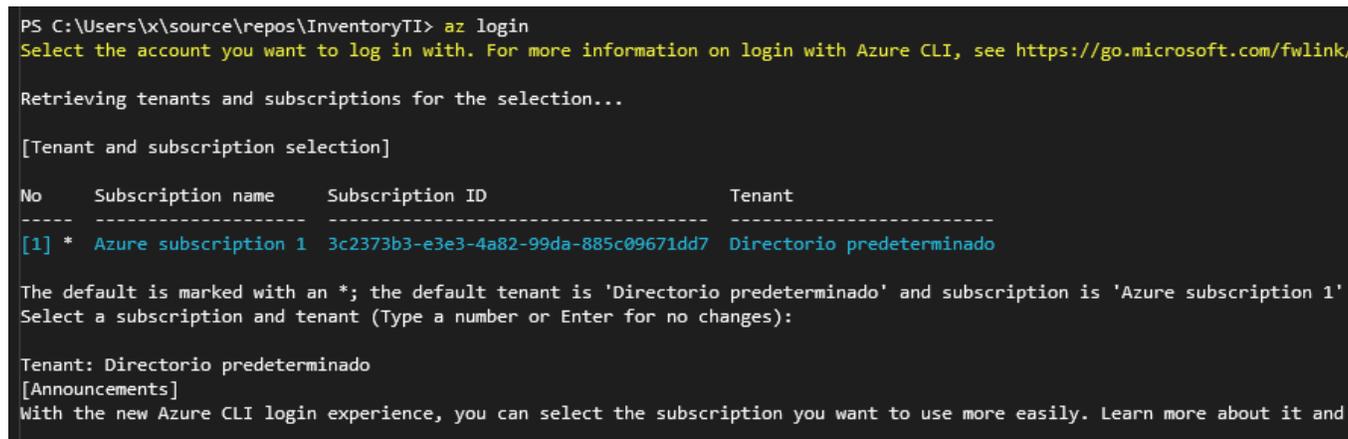
```
Administrador Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.6093]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>winget install --exact --id Microsoft.AzureCLI --version 2.67.0
El origen 'mstore' requiere que vea los siguientes contratos antes de usarlo.
Términos of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
El origen requiere que la región geográfica de dos letras de la máquina actual se envíe al servicio back-end para que funcione correctamente (por ejemplo, "EE. UU.>").
¿Está de acuerdo con todos los términos de los contratos de origen?
[Y] Sí [N] No: y
Encontrado Microsoft Azure CLI [Microsoft.AzureCLI] Versión 2.67.0
El propietario de esta aplicación le concede una licencia.
Microsoft no es responsable, ni tampoco concede ninguna licencia de paquetes de terceros.
Descargando https://azcliproduct.blob.core.windows.net/msi/azure-cli-2.67.0-x64.msi
[Progress bar] 67.8 MB / 67.8 MB
El hash del instalador se verificó correctamente.
Iniciando instalación de paquete...
Instalado correctamente
Notas: Winget installs the 64-bit CLI on 64-bit OS by default now. If you have used the 32-bit CLI before, please follow this guide to migrate to 64-bit version: https://learn.microsoft.com/cli/azure/install-azure-cli-windows#migrate-to-64-bit-azure-cli
C:\Windows\system32>
```

Con el comando

```
az login
```

Estaremos viendo que nos pedirá nuestra cuenta y suscripción activa, bastará con darle enter para poder recuperar la asociada a esa cuenta seleccionada:



```
PS C:\Users\x\source\repos\InventoryTI> az login
Select the account you want to log in with. For more information on login with Azure CLI, see https://go.microsoft.com/fwlink/?linkid=2148267

Retrieving tenants and subscriptions for the selection...

[Tenant and subscription selection]

No      Subscription name      Subscription ID      Tenant
-----
[1] *   Azure subscription 1  3c2373b3-e3e3-4a82-99da-885c09671dd7  Directorio predeterminado

The default is marked with an *; the default tenant is 'Directorio predeterminado' and subscription is 'Azure subscription 1'
Select a subscription and tenant (Type a number or Enter for no changes):

Tenant: Directorio predeterminado
[Announcements]
With the new Azure CLI login experience, you can select the subscription you want to use more easily. Learn more about it and
```

Iniciaremos sesión ahora con nuestro contenedor para la imagen mediante el comando

```
az acr login --name inventorytiapp
```

Eso nos va a situar a nivel del contenedor y todo lo que hagamos en la línea de comandos se indexará de manera automática a este mismo, por lo que una vez logueados deberemos generar la imagen de nuestro proyecto haciendo uso de:

```
docker build --no-cache -t inventoryti-backend:latest .
```

```
PS C:\Users\x\source\repos\InventoryTI> docker build --no-cache -t inventoryti-backend:latest .
[+] Building 54.1s (18/18) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 1.34kB
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0
=> [internal] load .dockerignore
=> => transferring context: 464B
=> [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:8.0@sha256:45e41fe52eb60f42bd75c83b7e8bfff0523e031e042b4c1fc7ddb9c348898c64
=> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:8.0@sha256:6d8901fae2d8f4c0e73962046a67fbdaf017bc8a6833c74ae87e9fbd0810b73
=> [internal] load build context
=> => transferring context: 1.99kB
=> CACHED [build 2/7] WORKDIR /src
=> CACHED [base 2/2] WORKDIR /app
=> CACHED [final 1/2] WORKDIR /app
=> [build 3/7] COPY [InventoryTI.csproj, .]
=> [build 4/7] RUN dotnet restore "./InventoryTI.csproj"
=> [build 5/7] COPY . .
=> [build 6/7] WORKDIR /src/.
=> [build 7/7] RUN dotnet build "./InventoryTI.csproj" -c Release -o /app/build
=> [publish 1/1] RUN dotnet publish "./InventoryTI.csproj" -c Release -o /app/publish /p:UseAppHost=false
=> [final 2/2] COPY --from=publish /app/publish .
=> exporting to image
=> => exporting layers
=> => writing image sha256:9d129aec42ec2b6c26132b5608f5e583531a82213978e24e864253023a74fda6
=> => naming to docker.io/library/inventoryti-backend:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/d62n5tdmnc7e7z3z53hrmur3
```

Con esto creamos una instancia de nuestra imagen y lo que seguiría es etiquetar nuestra imagen con dirección a nuestro acr, haciendo uso del dominio que se nos proporcionó al momento de crear nuestro contenedor

```
docker tag inventoryti-backend:latest inventorytiapp.azurecr.io/inventoryti-backend:latest
```

Para finalmente poder hacer Push a este Contenedor con el comando:

```
docker push inventorytiapp.azurecr.io/inventoryti-backend:latest
```

```
PS C:\Users\x\source\repos\InventoryTI> docker tag inventorytiapp.azurecr.io/inventoryti-backend:latest
>>
The push refers to repository [inventorytiapp.azurecr.io/inventoryti-backend]
32ce8f039dc7: Pushed
5f70bf18a086: Pushed
81b4fbe3211c: Pushed
d4d6422e4dc5: Pushed
6ee99d30c41c: Pushed
3e3a217c0576: Pushed
08f5a23a0fb0: Pushed
0bf603c4b9a2: Pushed
7cc7fe68eff6: Pushed
latest: digest: sha256:49538feede52a340866238bba162c00c860eebbb97afdc382cd97a2d5f62d size: 2203
PS C:\Users\x\source\repos\InventoryTI>
```

## ¿Cambios Nuevos?

Cada vez que quieras actualizar tu contenedor en Azure, bastará con repetir este proceso: reconstruir - etiquetar - subir.

Esto mantendrá tu ACA actualizada con la última versión del backend.

## 8. Creando nuestro Azure Container App (ACA)

Una vez que tenemos nuestra imagen subida al Azure Container Registry (ACR), el siguiente paso es **crear nuestra Azure Container App (ACA)**, que será la instancia encargada de ejecutar dicha imagen como backend.

Por lo que ahora tenemos dos maneras de crear nuestro ACA

### 8.1 Mediante la CLI

Podemos crear la ACA directamente con un comando como el siguiente:

```
az containerapp create --name inventory-api --resource-group Inventory-TI --environment inventory-env --image inventorytiapp.azurecr.io/inventoryti-backend:latest --target-port 8081 --ingress external --registry-server inventorytiapp.azurecr.io --cpu 0.5 --memory 1.0Gi --env-vars ASPNETCORE_ENVIRONMENT=Production
```

#### 🔗 Configuraciones

- Expone el puerto 8081 al público.
- Usa 0.5 vCPU y 1 GB RAM.
- Configura `ASPNETCORE_ENVIRONMENT` como `Production`.

### 8.2 Mediante la GUI



En caso de que escojamos la opción de IU llenaríamos cómo:

## Create Container app ...

Basics Container Ingress Tags Review + create

Create a containerized app and run it on a serverless platform—without managing cloud infrastructure. [Quickstart guide](#)

### Project details

Select a subscription to manage resource creation and costs, and a resource group to organize all your resources for this deployment.

Subscription \*

Resource group \*  [Create new resource group](#)

Container app name \*

Optimize for Azure Functions

Built-in support and autoscaling for Azure Functions (requires image compatible with Functions). [How to run functions with your container app](#)

Deployment source \*

Container image  
Bring your own container registry or build a container from a Dockerfile.

Source code or artifact  
Build and deploy your code without using a Dockerfile.

### Container Apps environment

An environment is a secure boundary around a group of container apps. [Container Apps Pricing](#)

Show environments in all regions ⓘ

Container Apps environment \*

---

## Contenedor

Ahora vamos a seleccionar toda la información de nuestro propio contenedor, aquí especificamos directamente que registro usaremos, la imagen, la etiqueta o tag que le hayamos colocado, en ambiente de desarrollo (.NET en nuestro caso) y los perfiles de carga de trabajo, cómo la capacidad de CPU y Memoria

## Create Container app ...

### Container details

|                |   |
|----------------|---|
| Name *         | <input type="text" value="inventory-api"/>  |
| Image source   | <input checked="" type="radio"/> Azure Container Registry<br><input type="radio"/> Docker Hub or other registries |
| Subscription * | <input type="text" value="Azure subscription 1"/>   |
| Registry *     | <input type="text" value="inventorytiapp.azurecr.io"/>  |
| Image *        | <input type="text" value="inventoryti-backend"/>  |
| Image tag *    | <input type="text" value="latest"/>   |

### Registry authentication

|                     |  |
|---------------------|--|
| Authentication type | <input type="radio"/> Managed identity <input checked="" type="radio"/> Secrets<br><b>i</b> Identity auth is only available for new environments or environments with existing system identities |
|---------------------|--|

|                    |   |
|--------------------|---|
| Command override ⓘ | <input type="text" value="Example: /bin/bash"/> |
|--------------------|---|

|                      |   |
|----------------------|---|
| Arguments override ⓘ | <input type="text" value="Example: -c, while true; do echo hello; sleep 10; done"/> |
|----------------------|---|

### Development stack-specific features

When you select a specific development stack, you get additional features tailored to that stack—optimizing Container Apps to perform for your unique settings.

|                   |                                   |
|-------------------|-----------------------------------|
| Development stack | <input type="text" value=".NET"/> |
|-------------------|-----------------------------------|

[Customize .NET features for your app](#)

**i** [Learn more](#) about .NET-specific features

### Container resource allocation

Choose the workload profile for this app. You can adjust the CPU and memory allocation for this app up to the workload profile limit. [Learn more](#)

|                    |  |
|--------------------|--|
| Workload profile * | <input type="text" value="Consumption - Up to 4 vCPUs, 8 Gib memory"/> |
|--------------------|--|

|                  |   |
|------------------|---|
| CPU and memory * | <input type="text" value="1 CPU cores, 2 Gi memory"/> |
|------------------|---|

---

## Entrada

En este apartado vamos a configurar las entradas a la aplicación, o los puertos por los cuáles entraran los usuarios, depende de nuestro entorno es como lo vamos a configurar, en este caso como estamos esperando que se conecte a swagger y entornos de desarrollo, podemos dejarlo expuesto a tráfico abierto, siempre y cuándo sean conexiones https:

## Create Container app ...

Basics Container Ingress Tags Review + create

### Application ingress settings

Enable ingress for applications that need an HTTP or TCP endpoint.

|                      |   |
|----------------------|---|
| Ingress ⓘ            | <input checked="" type="checkbox"/> Enabled   |
| Ingress traffic      | <input type="radio"/> Limited to Container Apps Environment<br>Select this option if you want to restrict traffic to this container app from within the Container App Environment |
|                      | <input checked="" type="radio"/> Accepting traffic from anywhere<br>Select this option if you want to allow traffic to this container app from anywhere                           |
| Ingress type ⓘ       | <input checked="" type="radio"/> HTTP   |
|                      | <input type="radio"/> TCP   |
| Transport            | <input type="text" value="Auto"/>   |
| Insecure connections | <input type="checkbox"/> Allowed  |
| Target port ⓘ        | <input type="text" value="8081"/>   |
| Session affinity ⓘ   | <input type="checkbox"/> Enabled  |

---

^ Additional TCP ports

## 9. Usando nuestro ACA

A continuación se presentan dos opciones para poder usar nuestro ACA de manera correcta teniendo en cuenta algunas configuraciones adicionales que tenemos que manejar

### Configuraciones adicionales

Con esta configuración damos permisos a la instancia para iniciar como administrador, para eso nos vamos a situar en **Configuración/Claves de acceso** y vamos a generar un usuario y contraseña, sin embargo esta configuración puede llegar a exponer nuestra aplicación

Buscar

- Información general
- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Inicio rápido
- Visualizador de recursos
- Eventos
- Configuración
  - Claves de acceso
  - Cifrado

Nombre del Registro: InventoryTIApp

Servidor de inicio de sesión: inventorytiapp.azurecr.io

Usuario administrador:

Nombre de usuario: InventoryTIApp

| Nombre    | Contraseña | Regenerar |
|-----------|------------|-----------|
| password  | .....      | Mos...    |
| password2 | .....      | Mos...    |

## Configuraciones Adicionales

La segunda configuración sería habilitar nuestra instancia para Identidades Administradas, para eso nos situaremos en **Configuración/Identidad** aquí vamos a activar el estado y eso nos generará nuestro ID de objeto

Inicio > InventoryTIApp

InventoryTIApp | Identidad ☆ ...  
Container registry

Asignado por el sistema Usuario asignado

Una identidad administrada asignada por el sistema está restringida a una por recurso y está ligada al ciclo de vida de este recurso. Puede conceder permisos a la identidad administrada mediante el control de acceso basado en roles de Azure (RBAC de Azure). La identidad administrada se autentica con Microsoft Entra ID, de modo que no tiene que almacenar credenciales en el código.

Guardar Descartar Actualizar ¿Tiene algún comentario?

Estado  Desactivado  **Activado**

Id. de objeto (entidad de seguridad)

Permisos

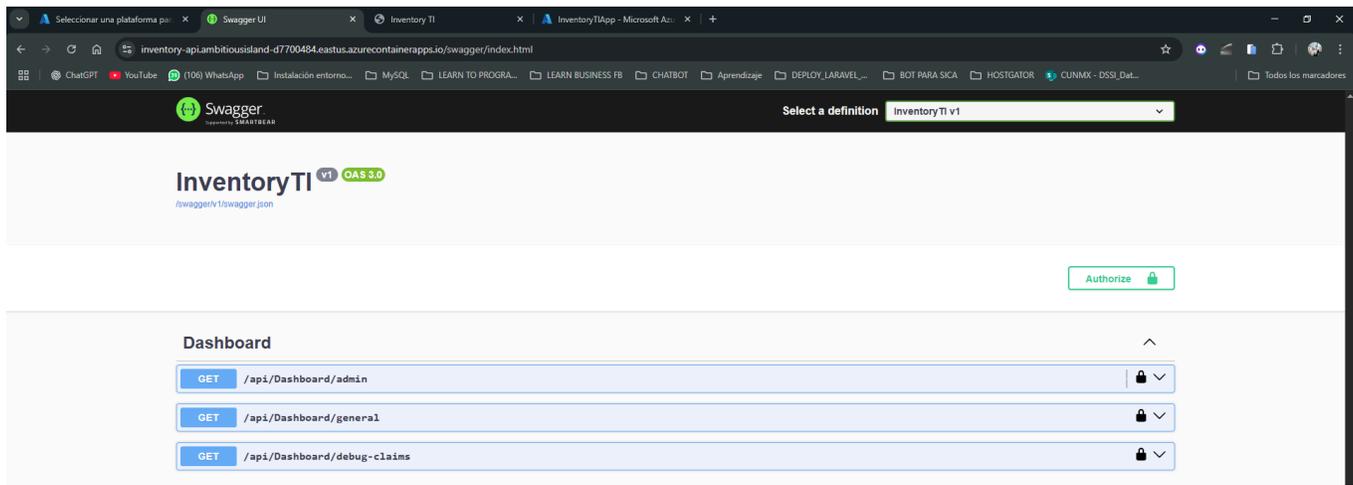
Este recurso se registró correctamente con Microsoft Entra ID. La identidad administrada se puede configurar para permitir el acceso a otros recursos. Tenga cuidado al realizar cambios en la configuración de acceso para la identidad administrada, ya que puede provocar errores. [Más información](#)

✔ Administración usuario desactivado  
'InventoryTIApp' está desactivado como administrador

Una vez que hayamos creado todo de manera correcta podemos acceder a nuestro ACA y recuperar la URL Pública de nuestra aplicación y probar el acceso al swagger.

| Essentials            |  | JSON View                 |  |
|-----------------------|--|---------------------------|--|
| Resource group (move) | : Inventory-TI                         | Application Url           | : https://inventory-api.agreeableriver-fa5dc220.centralus.azurecontaine... |
| Status                | : Running                              | Container Apps Environ... | : inventory-env  |
| Location (move)       | : Central US                           | Environment type          | : Workload profiles  |
| Subscription (move)   | : Azure subscription 1                 | Log Analytics             | : inventory-logs   |
| Subscription ID       | : 3c2373b3-e3e3-4a82-99da-885c09671dd7 | Development stack         | : .NET (manage)  |
| .NET Aspire Dashboard | : Not yet active (set up)              |                           |  |
| Tags (edit)           | : Add tags                             |                           |  |

Si todo ha funcionado de manera correcta deberíamos ser capaces de ver nuestro swagger corriendo.



## Creando nuestra VNet

Antes de pasar a crear nuestra base de datos es necesario que configuremos nuestras redes virtuales para poder hacer una conexión segura sin necesidad de exponer una IP estática, por lo que entre estos pasos adicionales debemos crear nuestra red virtual.

## Datos básicos

En este apartado vamos a seleccionar nuestra suscripción, el grupo al que pertenecerá nuestra red y colocar detalles de instancia como el nombre de la VNet y la región que estaremos usando:

# Crear red virtual ...

[Datos básicos](#) [Seguridad](#) [Direcciones IP](#) [Etiquetas](#) [Revisar y crear](#)

Azure Virtual Network (VNet) es el bloque de creación fundamental de su red privada en Azure. VNet habilita muchos tipos de recursos de Azure, como Azure Virtual Machines (VM), para comunicarse de forma segura entre sí, en Internet y en las redes locales. VNet es similar a una red tradicional que funciona en su propio centro de datos, pero ofrece ventajas adicionales de la infraestructura de Azure, como el escalado, la disponibilidad y el aislamiento.

[Más información.](#)

## Detalles del proyecto

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

Suscripción \*

Grupo de recursos \*

[Crear nuevo](#)

## Detalles de instancia

Nombre de red virtual \*

Región \*

[Implementación en una zona extendida de Azure](#)

## Seguridad

En este apartado vamos a seleccionar las protecciones que queremos agregar a nuestra red virtual para poder hacerla más segura y robusta, depende del giro de nuestra empresa serán las opciones que estaremos marcando, por ejemplo:

Cifrado de Red Virtual (CRV): Cifra tráfico dentro de la VNet entre recursos y esto requiere aceleración de red

Azure Bastion (AB): Proporciona conectividad RDP/SSH segura a las máquinas virtuales a través de TLS. Cuando se conecta a través de Azure Bastion, las máquinas virtuales no necesitan una dirección IP pública.

Azure Firewall (AF): Protege los recursos de Azure Virtual Network.

Protección de DDoS: Mitiga ataques DDoS.

# Crear red virtual ...

Datos básicos Seguridad Direcciones IP Etiquetas Revisar y crear

Mejore la seguridad de la red virtual con estos servicios de seguridad de pago adicionales. [Más información](#)

## Cifrado de red virtual

Habilita el cifrado de red virtual para cifrar el tráfico que viaja dentro de la red virtual. Las máquinas virtuales deben tener habilitadas las redes aceleradas. El tráfico a las direcciones IP públicas no está cifrado. [Más información](#)

Cifrado de red virtual

## Azure Bastion

Azure Bastion es un servicio de pago que proporciona conectividad RDP/SSH segura a las máquinas virtuales a través de TLS. Cuando se conecta a través de Azure Bastion, las máquinas virtuales no necesitan una dirección IP pública. [Más información](#)

Habilitar Azure Bastion

## Azure Firewall

Azure Firewall es un servicio de seguridad de redes administrado, basado en la nube, que protege los recursos de Azure Virtual Network. [Más información](#)

Habilitar Azure Firewall

## Protección de red Azure DDoS

La protección de red Azure DDoS es un servicio de pago que ofrece funcionalidades mejoradas de mitigación de DDoS mediante ajustes adaptables, notificación de ataques y telemetría para protegerse contra los impactos de un ataque DDoS para todos los recursos protegidos de esta red virtual. [Más información](#)

Habilitar la protección de red Azure DDoS

En caso de que estemos usando máquinas virtuales el CRV y AB serían indispensables.

## Direcciones IP

Aquí vamos a declarar el rango de redes IPv4 e IPv6 que vamos a estar manejando, para este apartado, si consideramos nuestro stack, la mejor opción de configuración sería una 10.0.0.0/16 por lo que podríamos hacer algo cómo:

Configure el espacio de direcciones de la red virtual con las direcciones IPv4 e IPv6 y las subredes que necesita. [Más información](#)

Defina el espacio de direcciones de la red virtual con uno o varios intervalos de direcciones IPv4 o IPv6. Cree subredes para segmentar el espacio de direcciones de la red virtual en intervalos más pequeños para que lo usen las aplicaciones. Al implementar recursos en una subred, Azure asigna al recurso una dirección IP de la subred. [Más información](#)

Asignar mediante grupos de direcciones IP. [Más información](#)

+ Agregar una subred

10.0.0.0/16 [Eliminar espacio de direcciones](#)

10.0.0.0 - 10.0.255.255 65.536 direcciones

| Subredes | Intervalo de direcciones IP | Tamaño                | NAT Gateway |
|----------|-----------------------------|-----------------------|-------------|
| default  | 10.0.0.0 - 10.0.0.255       | /24 (256 direcciones) | -           |

Agregar espacio de direcciones IPv4 | v

Para después configurar nuestras SubRedes:

1. sql-subnet
2. aca-subnet
3. acr-subnet

Estas tendrían las siguientes especificaciones

| Subred     | Rango       |
|------------|-------------|
| default    | 10.0.0.0/24 |
| sql-subnet | 10.0.1.0/24 |
| aca-subnet | 10.0.2.0/24 |
| acr-subnet | 10.0.3.0/24 |

De modo que al terminar la configuración deberíamos ver algo parecido a:

10.0.0.0/16 Eliminar espacio de direcciones

10.0.0.0 /16

10.0.0.0 - 10.0.255.255 65,536 direcciones

| Subredes   | Intervalo de direcciones IP | Tamaño                | NAT Gateway |   |
|------------|-----------------------------|-----------------------|-------------|---|
| default    | 10.0.0.0 - 10.0.0.255       | /24 (256 direcciones) | -           |   |
| sql-subnet | 10.0.1.0 - 10.0.1.255       | /24 (256 direcciones) | -           |   |
| aca-subnet | 10.0.2.0 - 10.0.2.255       | /24 (256 direcciones) | -           |   |
| acr-subnet | 10.0.3.0 - 10.0.3.255       | /24 (256 direcciones) | -           |   |

## Creando nuestro Azure SQL Database

En este punto el siguiente paso para construir nuestro entorno de azure es crear tanto el servidor de nuestra base de datos, así como esta misma.

Cuándo estemos llenando los campos necesarios para crear nuestro propio servidor veremos que nos pedirá datos cómo:

1. Nombre del servidor
2. Ubicación
3. Método de autenticación
4. Administrador de Microsoft Entra

En el caso del método de autenticación podremos escoger entre 3 opciones, sin embargo la más recomendable será la segunda

Método de autenticación

Usar la autenticación solo de Microsoft Entra  
 Uso de la autenticación de SQL y Microsoft Entra  
 Uso de la autenticación de SQL

Configurar administrador de Microsoft Entra \*

**No seleccionado**  
[Establecer administrador](#)

Inicio de sesión del administrador del servidor \*

Escribir inicio de sesión del administrador del servidor

Contraseña \*

Confirmar contraseña \*

Pues esta nos va a solicitar un usuario administrador, nuestro inicio de sesión (Usuario) y una contraseña, una vez que hayamos llenado estos campos podremos crear nuestro servidor.

## Crear un servidor de SQL Database

Microsoft

Nombre del servidor \*

inventory-server ✓  
 .database.windows.net

Ubicación \*

(US) East US ✓

**Autenticación**

**i** Azure Active Directory ahora es Microsoft Entra ID. [Más información](#) ↗

Seleccione los métodos de autenticación preferidos para acceder a este servidor. Cree un inicio de sesión de administrador servidor y una contraseña para acceder a su servidor con autenticación de SQL. Seleccione solo la autenticación de Microsoft Entra. [Más información](#) ↗ usar un usuario, grupo o aplicación de Microsoft Entra existente como administrador de Microsoft Entra [Más información](#) ↗ , o seleccione tanto la autenticación SQL como la de Microsoft Entra.

Método de autenticación

Usar la autenticación solo de Microsoft Entra  
 Uso de la autenticación de SQL y Microsoft Entra  
 Uso de la autenticación de SQL

Configurar administrador de Microsoft Entra

**guillermo.jesus.garcia.canul\_outlook.com#EXT#@guillermojesusgarciacanu756.onmicrosoft.com**  
 Id. de Objeto/Aplicación de la administración: 68d24f54-3ff2-48da-9a44-e43028794450  
[Establecer administrador](#)

Inicio de sesión del administrador del servidor \*

..... ✓

Contraseña \*

..... ✓

Confirmar contraseña \*

..... ✓

# Básico

Aquí podremos seleccionar directamente la configuración de nuestro servidor para la base de datos, los precios varían principalmente según las especificaciones que le agreguemos a este mismo, en caso de servidores dedicados, los precios podrían tomarse por segundo, por lo que para entornos de desarrollo podemos empezar con un flujo de implementación y para producción calibrar el servidor con base a nuestros requerimientos.

[Inicio](#) > [Bases de datos SQL](#) >

## Crear base de datos SQL

Microsoft

Servidor \* ⓘ  [Crear nuevo](#)

¿Quiere usar un grupo elástico de SQL? ⓘ  Sí  No

Entorno de carga de trabajo  Implementación  Producción

**i** Configuración predeterminada proporcionada para las cargas de trabajo de Development. Las configuraciones se pueden modificar según sea necesario.

Proceso y almacenamiento \* ⓘ

**Uso general - Sin servidor**  
Serie estándar (Gen 5), 1 vCore, Almacenamiento: 32 GB, redundancia de zona deshabilitada  
[Configurar base de datos](#)

### Redundancia del almacenamiento de copias de seguridad

Elija el modo de replicación de las copias de seguridad de PITR y LTR. La restauración geográfica o la posibilidad de recuperación tras una interrupción regional solo están disponibles si se ha seleccionado el almacenamiento con redundancia geográfica.

Redundancia de almacenamiento de copia de seguridad ⓘ  Almacenamiento de copias de seguridad con redundancia local  Almacenamiento de copias de seguridad con redundancia de zona  Almacenamiento de copias de seguridad con redundancia geográfica  Almacenamiento de copia de seguridad con redundancia de zona geográfica

En cuánto a los precios lo podremos visualizar de lado derecho mediante un resumen.



### Resumen del costo

#### Uso general (GP\_S\_Gen5\_1)

Costo por GB (en USD) 0.14

Almacenamiento máximo seleccionado (en GB) x 41.6

**COSTO DE ALMACENAMIENTO ESTIMADO** 5.74  
**POR MES** USD

**COSTO DE PROCESO POR NÚCLEO** 0.000174  
**VIRTUAL POR SEGUNDO <sup>1</sup>** USD

#### NOTAS

<sup>1</sup> Las bases de datos sin servidor se facturan en segundos de núcleo virtual en función de una combinación de uso de CPU y memoria. [Más información sobre la facturación sin servidor](#)

## Redes

En este apartado vamos a realizar la configuración necesaria para dejar privada nuestra conexión, esto gracias a nuestra VNet, por lo que vamos a rellenar los campos necesarios y asignar una DNS privada que será autogenerada:

## Crear un punto de conexión privado



Suscripción \* ⓘ Azure subscription 1

Grupo de recursos \* ⓘ Inventory-TI  
[Crear nuevo](#)

Ubicación \* East US

Nombre \* ⓘ inventory

Subrecurso de destino \* SqlServer

**Redes**

Para implementar el punto de conexión privado, seleccione una subred de la red virtual.  
[Más información sobre las redes de puntos de conexión privados](#)

Red virtual ⓘ Inventory (Inventory-TI)

Subred \* ⓘ sql-subnet

ⓘ Si tiene un grupo de seguridad de red (NSG) habilitado para la subred anterior, se deshabilitará para los puntos de conexión privados solo en esta subred. En el resto de recursos de la subred seguirá vigente el grupo de seguridad de red.

**Integración de DNS privado**

Para conectarse de forma privada con el punto de conexión privado, necesita un registro de DNS. Se recomienda integrar el punto de conexión privado en una zona DNS privada. También puede usar sus propios servidores DNS o crear registros de DNS con los archivos host de sus máquinas virtuales. [Más información sobre la integración de DNS privado](#)

Integrar con la zona DNS privada ⓘ  Sí  No

Zona DNS privada \* ⓘ (Nuevo) privatelink.database.windows.net

## Seguridad

En este campo y el de **Configuración adicional** veremos campos que podemos ajustar de acuerdo a nuestras necesidades específicas, por lo que depende directamente de nuestro entorno. Por esta razón, estas partes no serán cubiertas en esta documentación de manera guiada, pero sí daremos una breve descripción de para qué sirven:

- **Microsoft Defender para SQL:** Servicio adicional de protección avanzada que evalúa vulnerabilidades, detecta amenazas y genera alertas proactivas. Requiere un costo mensual adicional por servidor.
- **Libro de contabilidad:** Permite habilitar la verificación de integridad de los datos mediante un historial inmutable. Útil para entornos que requieren trazabilidad y cumplimiento normativo.
- **Identidad del servidor:** Permite asignar una identidad administrada (MSI) al servidor SQL para que interactúe de manera segura con otros servicios de Azure, como Azure Key Vault,

sin necesidad de credenciales.

- **Cifrado de datos transparente (TDE):** Se encarga de cifrar automáticamente los datos en reposo en la base de datos, incluyendo copias de seguridad. Puede ser administrado por Azure o de forma personalizada usando una clave propia en Key Vault.
- **Always Encrypted:** Una característica avanzada de seguridad que protege datos sensibles permitiendo que solo las aplicaciones autorizadas puedan acceder a ellos. Separa completamente los datos visibles del control del administrador de base de datos.

Básico Redes Seguridad Configuración adicional Etiquetas Revisar y crear

### Microsoft Defender para SQL

Proteja sus datos con Microsoft Defender para SQL, un paquete de seguridad unificada que incluye Evaluación de vulnerabilidad y Protección contra amenazas avanzada para su servidor. [Más información](#)

Empiece con una prueba gratuita de 30 días. Luego, el precio será de 15 USD al mes por servidor.

Habilitar Microsoft Defender para SQL \*  Iniciar prueba gratuita  
 Ahora no

### Libro de contabilidad

Los libros de contabilidad comprueban de forma concreta la integridad de los datos y detectan posibles manipulaciones. [Más información](#)

Libro de contabilidad **No configurado**  
[Configurar el libro de contabilidad](#)

### Identidad del servidor

Utilice identidades administradas asignadas por el sistema y asignadas por el usuario para permitir la administración de acceso central entre esta base de datos y otros recursos de Azure. [Más información](#)

Identidad del servidor **Sin habilitar**  
[Configurar identidades](#)

### Administración de claves de cifrado de datos transparente

La tecnología Cifrado de datos transparente cifra sus bases de datos, copias de seguridad y registros en reposo sin realizar cambios en la aplicación. Para habilitar el cifrado, vaya a cada base de datos. La configuración de nivel de base de datos, si está habilitada, invalidará la configuración de nivel de servidor. [Más información](#)

Clave de nivel de servidor  **Clave administrada por el servicio seleccionada**  
[Configuración del cifrado de datos transparente](#)

Clave de nivel de base de datos  **No configurado**  
[Configuración del cifrado de datos transparente](#)

### Always Encrypted

Always Encrypted es una familia de características de protección de datos líderes del sector que proporcionan una separación entre las personas que son propietarias de los datos y pueden verlos, y las personas que los administran pero no deben tener acceso, los administradores de bases de datos locales, los operadores de bases de datos en la nube u otros usuarios con privilegios elevados pero no autorizados. [Más información](#)

Habilitar enclaves seguros   Activado  Desactivado

# Configuración adicional

Esta sección permite definir parámetros específicos para la base de datos relacionados con el origen de los datos, el tipo de intercalación y las ventanas de mantenimiento. Aunque no es obligatorio modificar estos campos en todos los escenarios, es importante comprender su propósito en caso de que se necesiten ajustes personalizados.

- **Origen de datos:** Permite seleccionar si la base de datos se crea desde cero, a partir de una copia de seguridad, o utilizando datos de muestra. Por defecto, la opción seleccionada es **"Ninguno"**, lo que genera una base de datos vacía.
- **Intercalación:** Define cómo se ordenan y comparan los datos dentro de la base de datos. Afecta reglas de acento, sensibilidad a mayúsculas y al idioma. La opción por defecto es `SQL_Latin1_General_CP1_CI_AS`, adecuada para la mayoría de los entornos en español o inglés. Esta opción no puede modificarse una vez creada la base de datos.
- **Ventana de mantenimiento:** Permite seleccionar un horario preferido para que Azure aplique actualizaciones de mantenimiento planificadas. Por defecto, se usa la ventana de `17:00 a 8:00`, lo cual minimiza interrupciones durante horarios laborales. Es posible dejar esta opción con su valor predeterminado o elegir una ventana específica.

Básico   Redes   Seguridad   Configuración adicional   Etiquetas   Revisar y crear

Personalice parámetros de configuración adicionales, incluidos los datos de muestra y la intercalación.

## Origen de datos

Empiece con una base de datos vacía, restaure otra a partir de una copia de seguridad, o bien seleccione datos de muestra para rellenar una nueva.

Usar datos existentes \*

Ninguno    Copia de seguridad    Muestra

## Intercalación de base de datos

La intercalación de la base de datos define las reglas que ordenan y comparan los datos, y no se puede cambiar después de crear la base de datos. La intercalación de la base de datos predeterminada es `SQL_Latin1_General_CP1_CI_AS`. [Más información](#) ⓘ

Intercalación \* ⓘ

[Buscar una intercalación](#)

## Ventana de mantenimiento

Seleccione una ventana de mantenimiento preferida en la lista desplegable. Durante el mantenimiento, las bases de datos permanecen disponibles, pero algunas actualizaciones pueden requerir una conmutación por error. La ventana de mantenimiento predeterminada del sistema (de 5 p. m. a 8 a. m.) limita la mayoría de las actividades a este horario, pero pueden producirse actualizaciones urgentes fuera de él. Para asegurarse de que todas las actualizaciones se producen solo durante la ventana de mantenimiento, seleccione una opción no predeterminada. [Más información](#) ⓘ

Ventana de mantenimiento

Si todo ha salido bien ya hemos creado nuestro servidor SQL y nuestra base de datos

Inicio > Microsoft.SQLDatabase.newDatabaseNewServer\_52b4c7f6253745d38939f | Información general

Implementación

Buscar

Eliminar Cancelar Volver a implementar Descargar Actualizar

Información general

Entradas Salidas Plantilla

\*\*\* La implementación está en curso

Nombre de implementación : Microsoft.SQLDatabase.newDatabaseNewServer\_52b4c7f6253745d38939f  
 Suscripción : Azure subscription 1  
 Grupo de recursos : Inventory-TI

Hora de inicio : 6/8/2025, 11:38:25 a.m.  
 ID de correlación : d51c1a0f-0a67-4223-8039-f902b40c0d4

▼ Detalles de implementación

| Recurso  | Tipo                            | Estado   | Detalles de la operación |
|--|---------------------------------|----------|--------------------------|
| inventory-server/inventory                             | Microsoft.Sql/servers/databases | Accepted | Detalles de la operación |
| inventory-server/Default                               | Microsoft.Sql/servers/connectio | OK       | Detalles de la operación |
| inventory-server                                       | Microsoft.Sql/servers           | Created  | Detalles de la operación |
| SubnetPolicies-pe-b4771d15-fa39-405f-a40a-04f3ad543039 | Implementación                  | OK       | Detalles de la operación |

## Conectando nuestro ACA a SQL

Es importante mencionar que si nuestro ACA cuenta con una suscripción estándar no podremos hacer la conexión de nuestras redes, solamente lo podremos hacer en la versión premium, por lo que en caso de que la suscripción sea inferior deberemos actualizarla

Una vez que hayamos hecho la actualización de nuestro servicio podremos empezar a crear nuestro punto de conexión privado, de esta manera podremos hacer una perimetría punto a punto que evite conexiones no deseadas

Inicio > Container registries > InventoryTIApp | Redes >

### Crear un punto de conexión privado

1 Datos básicos 2 Recurso 3 Virtual Network 4 DNS 5 Etiquetas 6 Revisar y crear

Use los puntos de conexión privados para conectarse a un servicio o recurso de forma privada. El punto de conexión privado debe estar en la misma región que la red virtual, pero puede estar en otra región distinta de la del recurso de vínculo privado al que se está conectando. [Más información](#)

#### Detalles del proyecto

Suscripción \*

Grupo de recursos \*   
[Crear nuevo](#)

#### Detalles de instancia

Nombre \*

Nombre de la interfaz de red \*

Región \*

Con respecto a **Datos básicos**, **Recurso** estos se llenarán de forma automática, lo que nosotros debemos rellenar es:

# Virtual Network

Aquí vamos a escoger la VNet ACR que creamos y vamos a asignar un grupo de seguridad para mantener el control de los permisos que puedan tener los usuarios que entren a este entorno.

## Crear un punto de conexión privado

✓ Datos básicos   ✓ Recurso   **3 Virtual Network**   4 DNS   5 Etiquetas   6 Revisar y crear

### Redes

Para implementar el punto de conexión privado, seleccione una subred de la red virtual. [Más información](#)

Red virtual

Subred \*

Directiva de red para puntos de conexión privados   Deshabilitado [\(editar\)](#)

### Configuración de IP privada

Asignar dinámicamente la dirección IP

Asignar estáticamente la dirección IP

### Grupo de seguridad de la aplicación

Configure la seguridad de red como una extensión natural de la estructura de una aplicación. ASG le permite agrupar máquinas virtuales y definir directivas de seguridad de red basadas en esos grupos. Puede especificar un grupo de seguridad de la aplicación como origen o destino en una regla de seguridad de NSG [Más información](#)

[+ Crear](#)

Grupo de seguridad de la aplicación

## DNS

Aquí vamos a seleccionar por defecto la configuración que nos da Azure para que se asigne de manera automática

# Crear un punto de conexión privado

✓ Datos básicos ✓ Recurso ✓ Virtual Network **4 DNS** 5 Etiquetas 6 Revisar y crear

## Integración de DNS privado

Para conectarse de forma privada con el punto de conexión privado, necesita un registro de DNS. Se recomienda integrar el punto de conexión privado en una zona DNS privada. También puede usar sus propios servidores DNS o crear registros de DNS con los archivos host de sus máquinas virtuales. [Más información](#)

Integrar con la zona DNS privada  Sí  No

| Nombre de confi...     | Suscripción        | Grupo de recursos | Zona DNS privada               |
|------------------------|--------------------|-------------------|--------------------------------|
| privatelink-azurecr-io | Azure subscript... | Inventory-TI      | (nuevo) privatelink.azurecr.io |

Una vez configurado el proceso para el Endpoint Privado veremos su implementación completa:

Nombre de implementación : Microsoft.PrivateEndpoint-20250806120236  
Suscripción : Azure subscription 1  
Grupo de recursos : Inventory-TI

Hora de inicio : 6/8/2025, 12:14:36 p.m.  
Id. de correlación : 15c85f59-9ba5-413b-8797-b9a568d5032b

▼ Detalles de implementación

| Recurso  | Tipo                      | Estado  | Detalles de la operación |
|--|---------------------------|---------|--------------------------|
| PrivateDns-20250806121516                                | Implementación            | Created | Detalles de la operación |
| Inventory-private  | Punto de conexión privado | OK      | Detalles de la operación |
| AddNewApplicationSecurityGroupsDeployment-20250806121515 | Implementación            | OK      | Detalles de la operación |

## Habilitando el acceso a nuestra Identidad Administrada

Ahora que tenemos activa la IA para nuestro ACA, se nos asigna un Id de seguridad, y este será el que nos dará los permisos necesarios para acceder a nuestro Servidor SQL, este método de autenticación nos permite omitir claves de acceso en nuestro código, pues de manera automática se enlaza con la autenticación de nuestro servicio.

inventory-api | Identidad

Asignado por el sistema Usuario asignado

Una identidad administrada asignada por el sistema está restringida a una por recurso y está ligada al ciclo de vida de este recurso. Puede conceder permisos a la identidad administrada mediante el control de acceso basado en roles de Azure (RBAC de Azure). La identidad administrada se autentica con Microsoft Entra ID, de modo que no tiene que almacenar credenciales en el código.

Guardar Descartar Actualizar ¿Tiene algún comentario?

Estado  Desactivado  Activado

Id. de objeto (entidad de seguridad) 508c63d7-5078-4d88-978d-a19352614156

Permisos Asignaciones de roles de Azure

Este recurso se registró correctamente con Microsoft Entra ID. La identidad administrada se puede configurar para permitir el acceso a otros recursos. Tenga cuidado al realizar cambios en la configuración de acceso para la identidad administrada, ya que puede provocar errores. [Más información](#)

## Asignando permisos a la base de datos

Una vez que nuestra Azure Container App tiene activa la Identidad del sistema, debemos **darle permisos para que pueda autenticar contra el SQL Server**. Para esto, haremos dos cosas:

Lo primero será dirigirnos a:

SQL Server/Control de acceso (IAM)/Agregar asignación de rol.

- Rol: Azure SQL DB Contributor (o uno más limitado si así lo requiere tu entorno, como *Reader* y *DataWriter*)
- Asignar acceso a: Identidad administrada
- Miembro: `inventory-api` (o el nombre que tenga tu container app)

Esto le da acceso a nivel infraestructura.

#### Adición de la asignación de roles ...

Rol **Miembros** Condiciones Revisión y asignación

Rol seleccionado App Compliance Automation Administrator

Asignar acceso a  Usuario, grupo o entidad de servicio  Identidad administrada

Miembros [+ Seleccionar miembros](#)

| Nombre          | Id. de objeto                        | Tipo  |   |
|-----------------|--------------------------------------|-------|---|
| InventoryAdmins | 23ccbcc1-e01e-4e46-a532-3bf006126cef | Grupo |  |

Description

Posterior a eso deberemos de asignar una configuración de Zona DNS dirigiendonos a:

Punto de conexión privado/Configuración de DNS

Y desde ahí agregamos el DNS privado:

## Agregar configuración de zona DNS



Suscripción \*

Azure subscription 1

Zona DNS privada \*

privatelink.database.windows.net  
grupo de recursos: inventory-ti

Grupo de zonas DNS \*

predeterminado

Nombre de la configuración \*

privatelink\_database\_windows\_net



## Automatización de empaquetado

En este apartado veremos cómo automatizar estos procesos, para que cuando nosotros queramos subir nuestros cambios a la rama de producción no tengamos que hacer de manera manual la creación de la imagen y el despliegue, pese a que son 3 comandos, es mejor saber que podemos automatizar esos procesos tan repetitivos, por lo que si nos ubicamos en:

ACA/Configuración/Implementación

Vamos a poder seleccionar el repositorio y la rama que queremos que esté al escucha, así cuando nuestro equipo haga `push` a esa rama en específico, nuestro flujo de trabajo se encargue de generar la nueva imagen y desplegar en producción.

Buscar

Actualizar Envíenos sus comentarios

- Información general
- Registro de actividad
- Control de acceso (IAM)
- Etiquetas
- Diagnosticar y solucionar problemas
- Visualizador de recursos
- Aplicación
  - Revisiones y réplicas
  - Contenedores
  - Escala
  - Volúmenes
- Configuración
  - Implementación**
  - Pila de desarrollo
  - Dapr

### Implementación continua Artefacto (versión preliminar)

Está implementando en la aplicación automáticamente con Acciones de GitHub. Si quiere cambiar la configuración de GitHub, desconecte la implementación continua.

Configure Acciones de GitHub para que compile e implemente automáticamente el código en la aplicación contenedora. Tenga en cuenta que cada implementación creará una nueva revisión.

#### Configuración de GitHub

|                             |                    |
|-----------------------------|--------------------|
| Conectado como              | GuillermoSM33      |
| Organización                | GuillermoSM33      |
| Repositorio                 | Inventory_TI_Azure |
| Rama                        | main               |
| Archivo de flujo de trabajo | .github/workflows  |

## Modificaciones

Por defecto se genera un flujo de trabajo que se encarga de lanzar nuestra nueva imagen a producción, sin embargo en caso de que necesitemos más configuraciones o algún añadido para DevOps podemos editar el archivo generado y agregar lo que necesitemos