

Las variables de entorno permiten configurar dinámicamente la aplicación o el script que ejecuta el contenedor. Puede usar Azure Portal, PowerShell o la CLI de Azure para establecer variables al crear el contenedor. Las variables de entorno protegidas evitan que se muestre información confidencial en la salida del contenedor.

Cree una instancia de Azure Cosmos DB y use variables de entorno para pasar la información de conexión a una instancia de contenedor de Azure. Una aplicación del contenedor usa las variables para leer y escribir datos desde Azure Cosmos DB. Cree una variable de entorno y una variable de entorno protegida para poder ver la diferencia entre ellas.

Implementación de Azure Cosmos DB

1. Al implementar Azure Cosmos DB, debe proporcionar un nombre de base de datos único. Con fines de aprendizaje, ejecute este comando desde Cloud Shell para crear una variable de Bash que contenga un nombre único:

```
COSMOS_DB_NAME=aci-cosmos-db-$(RANDOM)
```

2. Ejecute este comando `az cosmosdb create` para crear la instancia de Azure Cosmos DB:

```
COSMOS_DB_ENDPOINT=$(az cosmosdb create \
  --resource-group learn-deploy-aci-rg \
  --name $COSMOS_DB_NAME \
  --query documentEndpoint \
  --output tsv)
```

Este comando puede tardar unos minutos en completarse.

`$COSMOS_DB_NAME` especifica el nombre de la base de datos único. El comando imprime la dirección del punto de conexión para la base de datos. Aquí, el comando guarda la dirección en la variable Bash `COSMOS_DB_ENDPOINT`.

3. Ejecute `az cosmosdb keys list` para obtener la clave de conexión de Azure Cosmos DB y almacénela en una variable Bash llamada `COSMOS_DB_MASTERKEY`:

```
COSMOS_DB_MASTERKEY=$(az cosmosdb keys list \
  --resource-group learn-deploy-aci-rg \
  --name $COSMOS_DB_NAME \
  --query primaryMasterKey \
  --output tsv)
```

Implementación de un contenedor que funcione con la base de datos

Cree una instancia de contenedor de Azure que pueda leer y escribir registros en la instancia de Azure Cosmos DB.

Las variables de dos entornos que ha creado en la última parte, `COSMOS_DB_ENDPOINT` y `COSMOS_DB_MASTERKEY`, conservan los valores que necesita para conectarse a la instancia de Azure Cosmos DB.

1. Ejecute el siguiente comando `az container create` para crear el contenedor:

```
az container create \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo \
  --image mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb \
  --ip-address Public \
  --location eastus \
  --environment-variables \
    COSMOS_DB_ENDPOINT=$COSMOS_DB_ENDPOINT \
    COSMOS_DB_MASTERKEY=$COSMOS_DB_MASTERKEY
```

azuredocs/azure-vote-front:cosmosdb hace referencia a una imagen de contenedor que ejecuta una aplicación ficticia para votar.

Tenga en cuenta el argumento `--environment-variables`. Este argumento especifica variables de entorno que se pasan al contenedor cuando este se inicia. La imagen de contenedor está configurada para buscar estas variables de entorno. Use el nombre del punto de conexión de Azure Cosmos DB y su clave de conexión.

2. Ejecute el comando `az container show` para obtener la dirección IP pública del contenedor:

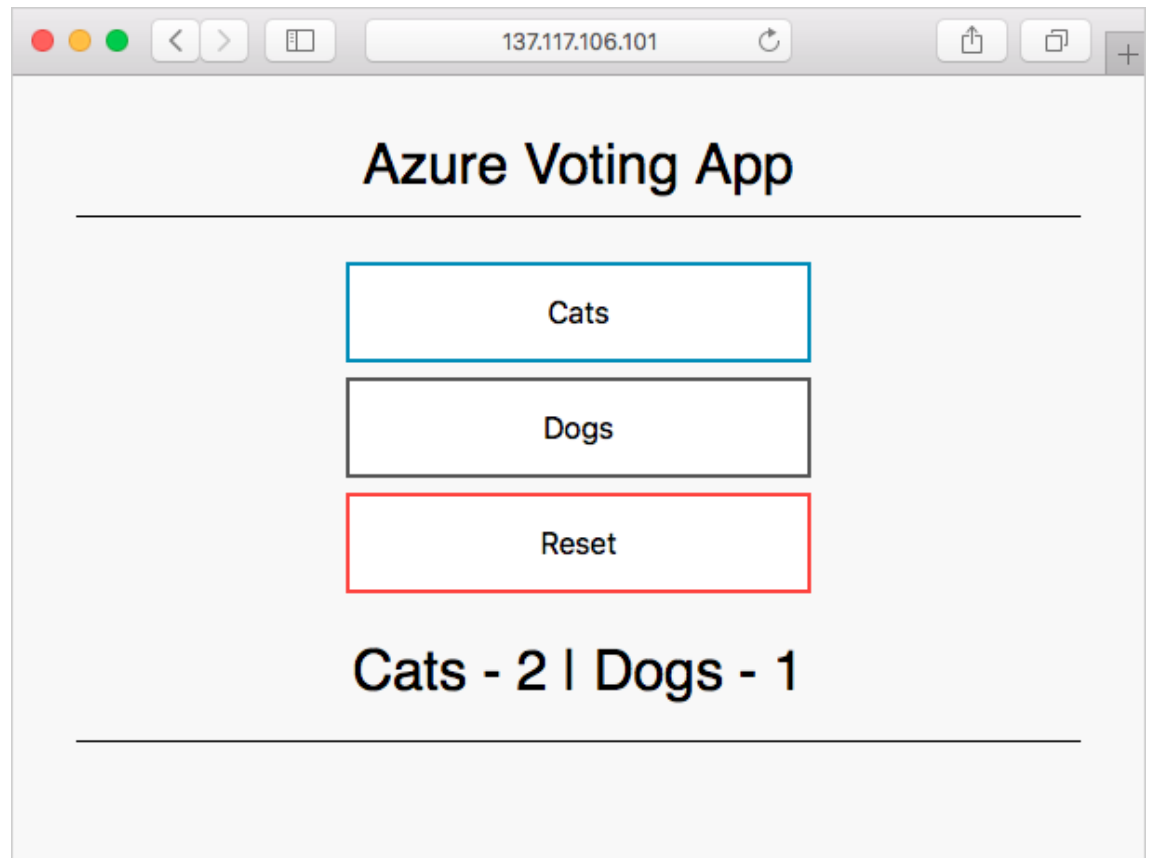
```
az container show \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo \
  --query ipAddress.ip \
  --output tsv
```

3. En un explorador, vaya a la dirección IP del contenedor.

Importante

En ocasiones, los contenedores tardan uno o dos minutos en iniciarse de forma completa y poder recibir conexiones. Si no hay ninguna respuesta al ir a la dirección IP en el explorador, espere unos minutos y actualice la página.

Una vez que la aplicación está disponible, se ve esta página:



Intente votar "Gatos" o "Perros". Cada voto se almacena en la instancia de Azure Cosmos DB.

Uso de variables de entorno protegido para ocultar la información de conexión

En la sección anterior, ha usado dos variables de entorno para crear el contenedor. De forma predeterminada, se puede acceder a las variables de entorno desde Azure Portal y en las herramientas de línea de comandos en texto sin formato.

En esta sección, se aprende a evitar que la información confidencial, como las claves de conexión, se muestren como texto sin formato.

1. Empecemos viendo el comportamiento actual en acción. Ejecute el comando `az container show` siguiente para mostrar las variables de entorno del contenedor:

```
az container show \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo \
  --query containers[0].environmentVariables
```

Se obtiene una salida con ambos valores en texto sin formato. Este es un ejemplo:

```
[
  {
    "name": "COSMOS_DB_ENDPOINT",
    "secureValue": null,
    "value": "https://aci-cosmos.documents.azure.com:443/"
  },
  {
    "name": "COSMOS_DB_MASTERKEY",
    "secureValue": null,
    "value":
      "Xm5BwdLlC1lBvrR26V0000000S2u0usuglhzwkE7d0PMBQ3oA30n3rKd8PKA137
      0000000095ynys863Ghgw=="
  }
]
```

Aunque estos valores no se muestren a los usuarios en la aplicación de votación, es una práctica de seguridad recomendada para garantizar que la información confidencial, como las claves de conexión, no se almacene como texto sin formato.

La protección de las variables de entorno evita la salida de texto no cifrado. Para usar variables de entorno protegido, debe utilizar el argumento `--secure-environment-variables`, en vez del argumento `-environment-variables`.

2. Ejecute el siguiente comando para crear otro contenedor denominado **aci-demo-secure** que use variables de entorno protegido:

```
az container create \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo-secure \
  --image mcr.microsoft.com/azuredocs/azure-vote-front:cosmosdb \
  --ip-address Public \
  --location eastus \
  --secure-environment-variables \
    COSMOS_DB_ENDPOINT=$COSMOS_DB_ENDPOINT \
    COSMOS_DB_MASTERKEY=$COSMOS_DB_MASTERKEY
```

Tenga en cuenta el uso del argumento `--secure-environment-variables`.

3. Ejecute el comando `az container show` siguiente para mostrar las variables de entorno del contenedor:

```
az container show \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo-secure \
  --query containers[0].environmentVariables
```

Ahora puede ver que las variables de entorno no aparecen en texto sin formato:

```
[
  {
    "name": "COSMOS_DB_ENDPOINT",
    "secureValue": null,
    "value": null
  },
  {
    "name": "COSMOS_DB_MASTERKEY",
    "secureValue": null,
    "value": null
  }
]
```

De hecho, los valores de estas variables no aparecen por ningún lado. Y está bien que así sea, porque estos valores hacen referencia a información confidencial. En este caso, todo lo que necesita saber es que las variables de entorno existen.

De forma predeterminada, Azure Container Instances no tiene estado. Si el contenedor se bloquea o se detiene, se pierde todo su estado. Para conservar el estado más allá de la vigencia del contenedor, debe montar un volumen desde un almacén externo.

Monte un recurso compartido de archivos de Azure en una instancia de contenedor de Azure para poder almacenar datos y acceder a ellos más adelante.

Creación de un recurso compartido de archivos de Azure

Cree una cuenta de almacenamiento y un recurso compartido de archivos. Posteriormente, se puede hacer que una instancia de contenedor de Azure acceda a este recurso.

1. La cuenta de almacenamiento necesita un nombre único. Con fines de aprendizaje, ejecute este comando para almacenar un nombre único en una variable de Bash:

```
STORAGE_ACCOUNT_NAME=mystorageaccount$RANDOM
```

2. Ejecute el siguiente comando `az storage account create` para crear la cuenta de almacenamiento:

```
az storage account create \
  --resource-group learn-deploy-aci-rg \
  --name $STORAGE_ACCOUNT_NAME \
  --sku Standard_LRS \
  --location eastus
```

3. Ejecute el siguiente comando para colocar la cadena de conexión de la cuenta de almacenamiento en una variable de entorno denominada `AZURE_STORAGE_CONNECTION_STRING`:

```
export AZURE_STORAGE_CONNECTION_STRING=$(az storage account show-
connection-string \
  --resource-group learn-deploy-aci-rg \
  --name $STORAGE_ACCOUNT_NAME \
  --output tsv)
```

AZURE_STORAGE_CONNECTION_STRING es una variable de entorno especial que la CLI de Azure comprende. El elemento `export` permite que otros comandos CLI que se ejecutan después accedan a la variable.

4. Ejecute este comando para crear un recurso compartido de archivos denominado **aci-share-demo** en la cuenta de almacenamiento:

```
az storage share create --name aci-share-demo
```

Obtención de credenciales de almacenamiento

Para montar un recurso compartido de archivos de Azure como un volumen en Azure Container Instances, necesita estos tres valores:

- Nombre de la cuenta de almacenamiento
- Nombre del recurso compartido
- Clave de acceso de cuenta de almacenamiento

Ya tiene los dos primeros valores. El nombre de cuenta de almacenamiento se almacena en la variable de Bash `STORAGE_ACCOUNT_NAME`. Se especificó **aci-share-demo** como el nombre del recurso compartido en el paso anterior. Aquí obtendrá el valor que falta: la clave de acceso de la cuenta de almacenamiento.

1. Ejecute el siguiente comando para obtener la clave de la cuenta de almacenamiento:

```
STORAGE_KEY=$(az storage account keys list \
  --resource-group learn-deploy-aci-rg \
  --account-name $STORAGE_ACCOUNT_NAME \
  --query "[0].value" \
  --output tsv)
```

El resultado se almacena en una variable de Bash llamada `STORAGE_KEY`.

2. Como paso opcional, imprima el nombre de la clave de almacenamiento en la consola.

```
echo $STORAGE_KEY
```

Implementación de un contenedor y montaje del recurso compartido de archivos

Para montar un recurso compartido de archivos de Azure como volumen en un contenedor, especifique el recurso compartido y el punto de montaje del volumen al crear el contenedor.

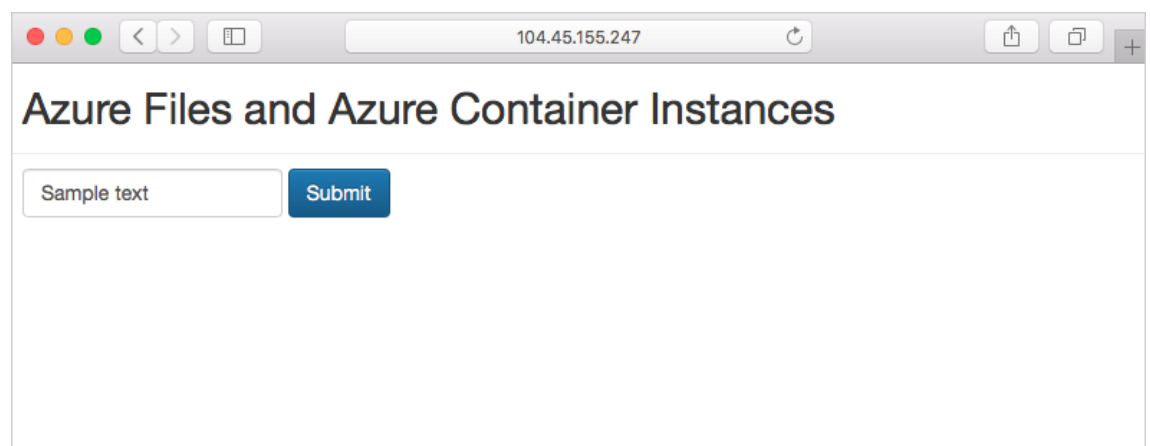
1. Ejecute este comando `az container create` para crear un contenedor que monte `/aci/logs/` en el recurso compartido de archivos:

```
az container create \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo-files \
  --image mcr.microsoft.com/azuredocs/aci-hellofiles \
  --location eastus \
  --ports 80 \
  --ip-address Public \
  --azure-file-volume-account-name $STORAGE_ACCOUNT_NAME \
  --azure-file-volume-account-key $STORAGE_KEY \
  --azure-file-volume-share-name aci-share-demo \
  --azure-file-volume-mount-path /aci/logs/
```

2. Ejecute `az container show` para obtener la dirección IP pública de su contenedor:

```
az container show \
  --resource-group learn-deploy-aci-rg \
  --name aci-demo-files \
  --query ipAddress.ip \
  --output tsv
```

3. Desde un explorador, vaya a la dirección IP de su contenedor. Aparece esta página:



4. Escriba cualquier texto en el formulario y seleccione **Enviar**. Esta acción crea un archivo que contiene el texto que escribió en el recurso compartido de archivos de Azure.
5. Ejecute este comando `az storage file list` para mostrar los archivos que se encuentran en el recurso compartido de archivos:


```
az storage file list -s aci-share-demo -o table
```

6. Ejecute `az storage file download` para descargar un archivo en la sesión de Cloud Shell. Reemplace **<filename>** por uno de los archivos que aparecían en el paso anterior.

```
az storage file download -s aci-share-demo -p <filename>
```

7. Ejecute el comando `cat` para imprimir el contenido del archivo.

```
cat <filename>
```

Recuerde que los datos se conservan cuando se cierra el contenedor. Puede montar el recurso compartido de archivos en otras instancias de contenedor para hacer que los datos estén disponibles.