

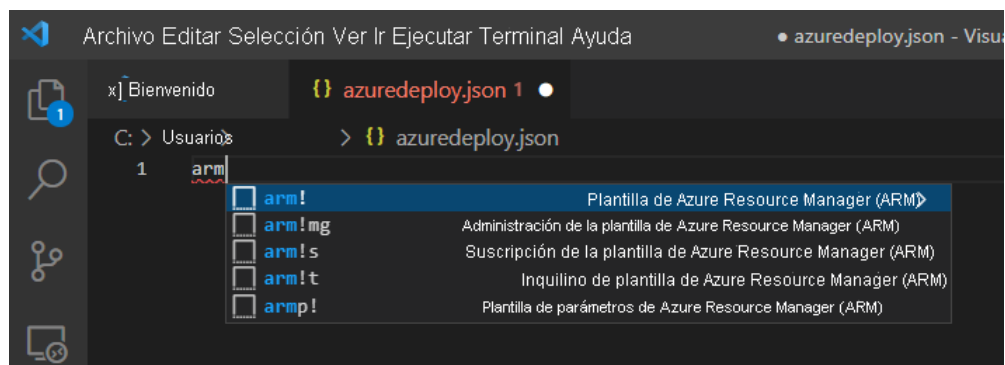
Creación e implementación de una plantilla de Azure Resource Manager

En este ejercicio, creará una plantilla de Azure Resource Manager (plantilla de ARM), la implementará en Azure y, a continuación, actualizará esa plantilla de ARM para agregar parámetros y salidas.

En este ejercicio se utilizan las [Herramientas de Azure Resource Manager para Visual Studio Code](#). Asegúrese de instalar esta extensión en Visual Studio Code antes de iniciar el ejercicio.

Creación de una plantilla de ARM

1. Abra Visual Studio Code y cree un archivo denominado *azuredeploy.json*.
2. La extensión de plantilla de ARM de Visual Studio Code viene configurada con fragmentos de código que le ayudarán a desarrollar plantillas. Para empezar, agregará una plantilla en blanco. En la primera línea del archivo, escriba *arm*.
3. Visual Studio Code muestra automáticamente varias opciones posibles que comienzan por **arm!**. Seleccione la **plantilla de Azure Resource Manager (ARM)**. Visual Studio Code procesa automáticamente los esquemas y lenguajes de la plantilla.



El archivo ahora tiene este aspecto:

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {},
  "functions": [],
  "variables": {},
  "resources": [],
```

```
"outputs": {}  
}
```

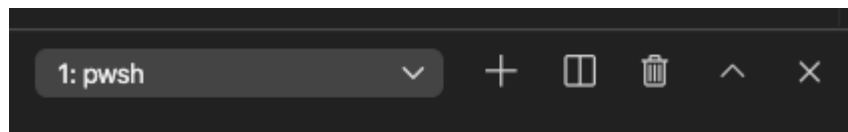
Observe que este archivo tiene todas las secciones de una plantilla de ARM que se describen en la unidad anterior.

4. Presione **Ctrl+S** para guardar los cambios en el archivo.

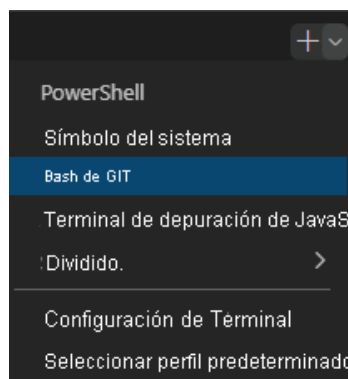
Implementación de la plantilla de ARM en Azure

Para implementar esta plantilla en Azure, debe iniciar sesión en la cuenta de Azure desde el terminal de Visual Studio Code. Asegúrese de haber instalado Azure PowerShell Tools desde las extensiones de Visual Studio Code e inicie sesión en la misma cuenta con la que ha activado el espacio aislado.

1. En la barra de comandos, seleccione **Terminal > Nuevo terminal** para abrir una ventana de PowerShell.
2. Si la barra de comandos de la ventana de terminal muestra **PowerShell**, tiene el shell correcto desde el que trabajar y puede ir a la sección siguiente.



- a. Si no es así, seleccione la flecha abajo y, en la lista desplegable, PowerShell. Si esa opción no aparece, seleccione **Select Default Profile** (Seleccionar el perfil predeterminado).
- b. En el campo de entrada, desplácese hacia abajo y seleccione **PowerShell**.



- c. Seleccione **Terminal > Nuevo terminal** para abrir una ventana de terminal de PowerShell.

Inicio de sesión en Azure mediante Azure PowerShell

1. Desde el terminal de Visual Studio Code, ejecute el comando siguiente para iniciar sesión en Azure. Se abre un explorador para que pueda iniciar sesión en su cuenta.

`Connect-AzAccount`

Sugerencia

El [módulo Az de PowerShell](#) reemplaza a AzureRM y es la versión recomendada que se debe usar para interactuar con Azure.

2. Inicie sesión con la cuenta que ha usado para activar el espacio aislado. Después de iniciar sesión, Visual Studio Code enumera las suscripciones asociadas a su cuenta en la ventana de terminal. Si ha activado el espacio aislado, verá un bloque de código que contiene "name": "Concierge Subscription". Esta es la suscripción que se usará para el resto del ejercicio.

Establecimiento de la suscripción predeterminada para todos los comandos de PowerShell de esta sesión

1. Ejecute el siguiente comando para obtener las suscripciones y sus id. El identificador de la suscripción es la segunda columna. Busque *Suscripción de Concierge* y copie el valor de la segunda columna. Es similar a `cf49fbbc-217c-4eb6-9eb5-a6a6c68295a0`:

`Get-AzSubscription`

2. Ejecute el siguiente comando y reemplace *{Your subscription ID}* por el id. que ha copiado en el paso anterior para cambiar la suscripción activa por la Suscripción de Concierge.

```
$context = Get-AzSubscription -SubscriptionId {Your subscription ID}
Set-AzContext $context
```

3. Ejecute el siguiente comando para permitir que el grupo de recursos predeterminado sea el grupo de recursos que se ha creado automáticamente en el entorno del espacio aislado. Esta acción le permite omitir ese parámetro del resto de los comandos de Azure PowerShell en este ejercicio.

`Set-AzDefault -ResourceGroupName [sandbox resource group name]`

Implementación de la plantilla en Azure

Implemente la plantilla en Azure mediante la ejecución de los comandos siguientes. La plantilla de ARM todavía no tiene ningún recurso, por lo que no verá recursos creados.

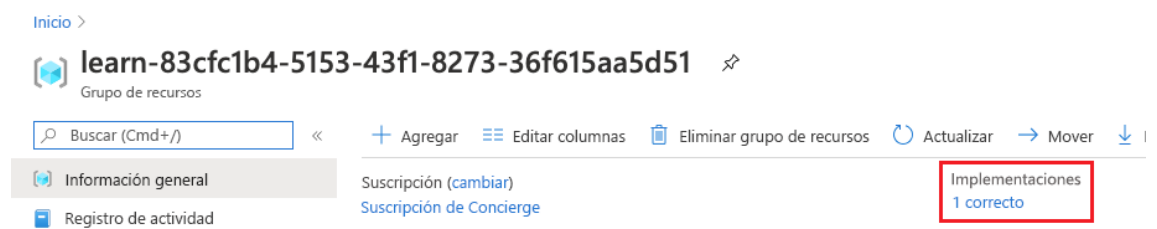
```
$templateFile="azuredeploy.json"
$today=Get-Date -Format "MM-dd-yyyy"
$deploymentName="blanktemplate-"+$today
New-AzResourceGroupDeployment `
  -Name $deploymentName `
  -TemplateFile $templateFile
```

La sección superior del código anterior establece las variables de Azure PowerShell, que incluyen la ruta de acceso al archivo de la implementación y el nombre de la implementación. Posteriormente, el comando `New-AzResourceGroupDeployment` implementa la plantilla en Azure. Observe que el nombre de la implementación es `blanktemplate` con la fecha como sufijo.

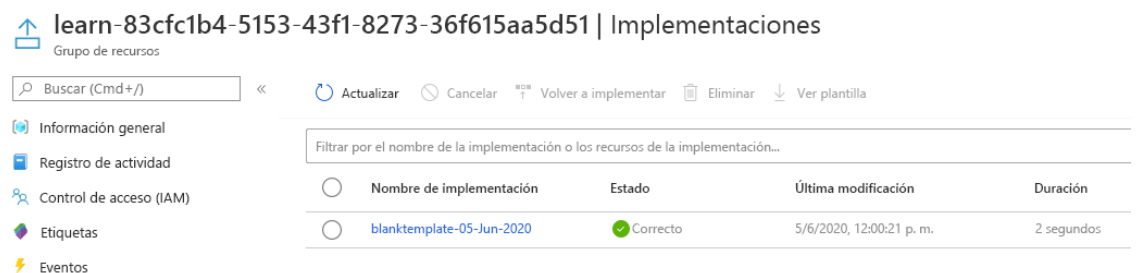
Cuando haya implementado la plantilla de ARM en Azure, vaya a [Azure Portal](#) y asegúrese de que se encuentra en la suscripción de espacio aislado. Para ello, seleccione el avatar en la esquina superior derecha de la página.

Seleccione **Cambiar directorio**. En la lista, seleccione el directorio **Espacio aislado de Microsoft Learn**.

1. En el menú de recursos, seleccione **Grupos de recursos**.
2. Seleccione el grupo de recursos *[nombre del grupo de recursos del espacio aislado]*.
3. En el panel **Información general**, verá la implementación realizada correctamente.



4. Seleccione **1 correcta** para ver los detalles de la implementación.



5. Seleccione `blanktemplate` para ver qué recursos se han implementado. En este caso, estará vacío porque todavía no ha especificado ningún recurso en la plantilla.

Inicio > learn-83cfc1b4-5153-43f1-8273-36f615aa5d51 | Implementaciones >

blanktemplate-05Jun-2020 | Overview

Buscar (Cmd+/) « Eliminar Cancelar Volver a implementar Actualizar

Información general

- Entradas
- Salidas
- Plantilla

Se completó la implementación

Nombre de implementación: blanktemplate-05-Jun-2020 Hora de inicio: 6/5/2020, 12:00:19 PM
Suscripción: Suscripción de Concierge Id. de correlación: 5a63dd04-65ef-4e5b-a3b6-efda59dcdr
Grupo de recursos: learn-83cfc1b4-5153-43f1-8273-36f615aa5d51

Detalles de implementación (Descargar)

| Recurso | Tipo | Estado | Detalles de la operación |
|-----------------|------|--------|--------------------------|
| Sin resultados. | | | |

6. Deje la página abierta en el explorador. Tendrá que volver a comprobar las implementaciones.

Adición de un recurso a la plantilla de ARM

En la tarea anterior, ha aprendido a crear una plantilla en blanco y a implementarla. Ahora está listo para implementar un recurso real. En esta tarea, agregará un recurso de cuenta de Azure Storage a la plantilla de ARM mediante un fragmento de código de la extensión Herramientas de Azure Resource Manager para Visual Studio Code.

1. En el archivo `azuredeploy.json` de Visual Studio Code, coloque el cursor entre los corchetes del bloque de recursos `"resources": [],`.
2. Escriba `storage` dentro de los corchetes. Aparece una lista de fragmentos de código relacionados. Seleccione **arm-storage**.

```
"parameters": {},  
"functions": [],  
"variables": {},  
"resources": [storage],  
"outputs": {}  
}
```

El archivo tendrá este aspecto:

```
JSONCopiar  
{
```

```

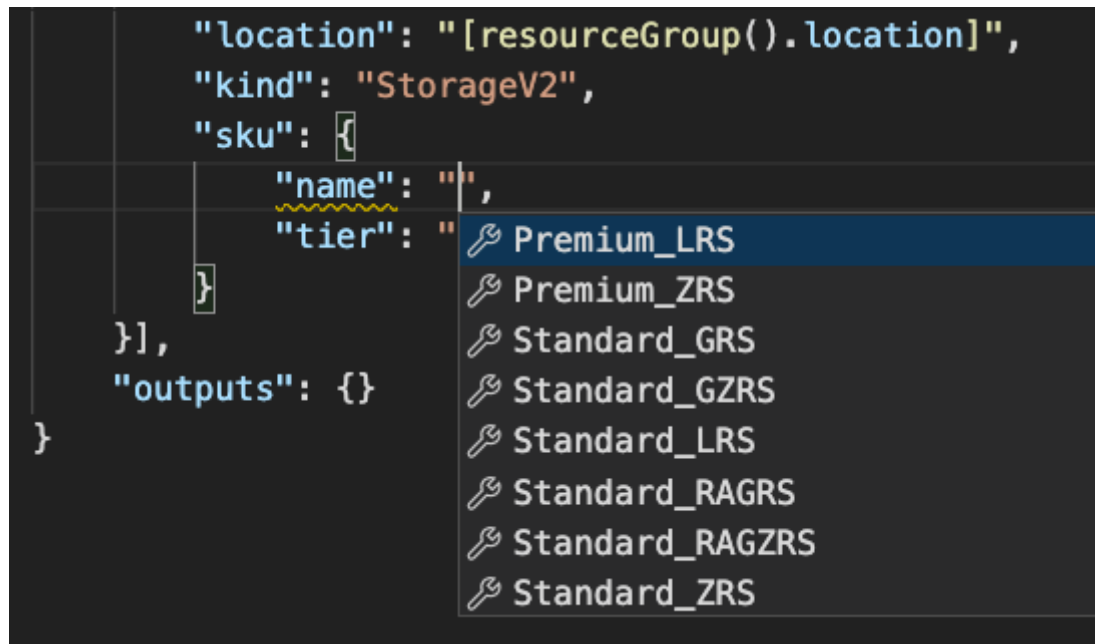
"$schema": "https://schema.management.azure.com/schemas/2019-
04-01/deploymentTemplate.json#",
"contentVersion": "1.0.0.0",
"parameters": {},
"functions": [],
"variables": {},
"resources": [
  {
    "name": "storageaccount1",
    "type": "Microsoft.Storage/storageAccounts",
    "apiVersion": "2019-06-01",
    "tags": {
      "displayName": "storageaccount1"
    },
    "location": "[resourceGroup().location]",
    "kind": "StorageV2",
    "sku": {
      "name": "Premium_LRS",
      "tier": "Premium"
    }
  }
],
"outputs": {}
}

```

Los valores que se deben editar se resaltan en la nueva sección del archivo y se puede navegar por ellos mediante la tecla Tabulador.

Observe que los atributos `tags` y `location` están rellenos. El atributo `location` usa una función para establecer la ubicación del recurso en la del grupo de recursos. Obtendrá información sobre etiquetas y funciones en el módulo siguiente.

3. Cambie los valores *name* y *displayName* del recurso por algo único (por ejemplo, **learnexercise12321**). Este nombre debe ser único en todo Azure, por lo que debe elegir algo que sea único para usted.
4. Cambie el valor *name* de sku de **Premium_LRS** a **Standard_LRS**. Cambie el valor de *tier* a **Estándar**. Observe que Visual Studio Code proporciona las opciones adecuadas para los valores de atributo en IntelliSense. Elimine el valor predeterminado, incluídas las comillas, y escriba las comillas para ver este trabajo.



5. La ubicación del recurso se establece en la del grupo de recursos donde se implementará. Mantenga este valor predeterminado.
6. Guarde el archivo.

Implementación de la plantilla de ARM actualizada

Aquí, cambiará el nombre de la implementación para que refleje mejor lo que hace.

Ejecute el siguiente comando de Azure PowerShell en el terminal. Este fragmento de código es el mismo código que ha usado antes, pero se ha cambiado el nombre de la implementación.

```
$templateFile="azuredeploy.json"
$today=Get-Date -Format "MM-dd-yyyy"
$deploymentName="addstorage-"+$today"
New-AzResourceGroupDeployment `
  -Name $deploymentName `
  -TemplateFile $templateFile
```

Comprobación de la implementación

1. Cuando finalice la implementación, vuelva al Azure Portal en el explorador. Vaya al grupo de recursos y comprobará que ahora hay **2 implementaciones correctas**. Seleccione este vínculo.

Observe que las dos implementaciones están en la lista.

Inicio >

learn-83cfc1b4-5153-43f1-8273-36f615aa5d51 | Implementaciones ✕

Grupo de recursos

Buscar (Cmd+/) << Actualizar Cancelar Volver a implementar Eliminar Ver plantilla

Información general

Registro de actividad

Control de acceso (IAM)

Etiquetas

Eventos

Filtrar por el nombre de la implementación o los recursos de la implementación...

| Nombre de la implementación | Estado | Última modificación | Duración | Eventos relacionados |
|--|------------|---------------------|------------|--------------------------------------|
| <input type="checkbox"/> addstorage-05-Jun-2020 | ✓ Correcto | 5/6/2020, 12:48:02 | 3 segundos | Eventos relacionados |
| <input type="checkbox"/> blanktemplate-05-Jun-2020 | ✓ Correcto | 5/6/2020, 12:00:21 | 2 segundos | Eventos relacionados |

2. Seleccione **addstorage**.

addstorage-05-Jun-2020 | Información general ✕

Implementación

Buscar (Cmd+/) << Eliminar Cancelar Volver a implementar Actualizar

Información general

Entradas

Salidas

Plantilla

✓ Se completó la implementación

Nombre de la implementación: addstorage-05-Jun-2020 Hora de inicio: 5/6/2020, 12:47:59
 Suscripción: [Suscripción de Concierge](#) Id. de correlación: a49fd90a-95aa-4ce8-b33e-292395e165
 Grupo de recursos: [learn-83cfc1b4-5153-43f1-8273-36f615aa5d51](#)

^ Detalles de implementación (Descargar)

| Recurso | Tipo | Estado | Detalles de la operación |
|--------------------|-----------------------------|---------|--|
| ✓ teststorage55555 | Microsoft.Storage/storag... | Aceptar | Detalles de la operación |

Observe que la cuenta de almacenamiento se ha implementado.