

Adición de parámetros y salidas a la plantilla de Azure Resource Manager

En este ejercicio, agregará un parámetro para definir el nombre de la cuenta de almacenamiento de Azure durante la implementación. Después, agregará un parámetro para definir qué SKU de cuenta de almacenamiento se permiten y definir cuál se va a usar para esta implementación. También agregará utilidad a la plantilla de Azure Resource Manager (plantilla de ARM) mediante la adición de una salida que se puede usar más adelante en el proceso de implementación.

Creación de parámetros para la plantilla de ARM

Aquí, puede hacer que la plantilla de ARM sea más flexible si agrega parámetros que se pueden establecer en tiempo de ejecución. Cree un parámetro para el valor `storageName`.

1. En el archivo `azuredeploy.json` de Visual Studio Code, coloque el cursor entre las llaves del atributo `parameters`. Su aspecto es similar a este: `"parameters": {}`,
2. Seleccione **Entrar** y luego **par**. Verá una lista de fragmentos de código relacionados. Elija **new-parameter**, que agrega un parámetro genérico a la plantilla. Tiene este aspecto:

```
"parameters": {  
  "parameter1": {  
    "type": "string",  
    "metadata": {  
      "description": "description"  
    }  
  }  
},
```

3. Cambie el parámetro de **parameter1** a **storageName** y deje el tipo como una cadena. Agregue un valor **minLength** de **3** y un valor **maxLength** de **24**. Agregue un valor de descripción de **Nombre del recurso de almacenamiento de Azure**.
4. El bloque de parámetros debería tener este aspecto:

```
"parameters": {  
  "storageName": {  
    "type": "string",  
    "minLength": 3,  
    "maxLength": 24,  
    "metadata": {
```

```

        "description": "The name of the Azure storage resource"
    }
}
},

```

5. Use el nuevo parámetro del bloque `resources` en los valores `name` y `displayName`. El archivo completo tendrá este aspecto:

```

{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "storageName": {
      "type": "string",
      "minLength": 3,
      "maxLength": 24,
      "metadata": {
        "description": "The name of the Azure storage resource"
      }
    }
  },
  "functions": [],
  "variables": {},
  "resources": [
    {
      "name": "[parameters('storageName')]",
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2019-06-01",
      "tags": {
        "displayName": "[parameters('storageName')]"
      },
      "location": "[resourceGroup().location]",
      "kind": "StorageV2",
      "sku": {
        "name": "Standard_LRS",
        "tier": "Standard"
      }
    }
  ],
  "outputs": {}
}

```

6. Guarde el archivo.

Implementación de la plantilla de ARM con parámetros

Aquí, cambiará el nombre de la implementación para que refleje mejor lo que hace y rellenará un valor para el nuevo parámetro.

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. Rellene un nombre único para el parámetro `storageName`. Recuerde que este nombre debe ser único en todo

Azure. Puede usar el nombre único que ha creado en la última unidad. En ese caso, Azure actualizará el recurso en lugar de crear uno.

```
$templateFile="azuredeploy.json"
$today=Get-Date -Format "MM-dd-yyyy"
$deploymentName="addnameparameter-"+$today
New-AzResourceGroupDeployment `
  -Name $deploymentName `
  -TemplateFile $templateFile `
  -StorageName {your-unique-name}
```

Comprobación de la implementación

1. Cuando finalice la implementación, vuelva al Azure Portal en el explorador. Vaya al grupo de recursos y compruebe que ahora hay **3 implementaciones correctas**. Seleccione este vínculo.

Observe que las tres implementaciones están en la lista.

2. Explore la implementación *addnameparameter* como ha hecho antes.

Adición de otro parámetro para limitar los valores permitidos

Aquí se usan parámetros para limitar los valores permitidos para un parámetro.

1. Coloque el cursor después de la llave de cierre del parámetro `storageName`. Agregue una coma y seleccione **Entrar**.
2. De nuevo, escriba **par** y seleccione **new-parameter**.
3. Cambie el nuevo parámetro genérico por lo siguiente:

```
"storageSKU": {
  "type": "string",
  "defaultValue": "Standard_LRS",
  "allowedValues": [
    "Standard_LRS",
    "Standard_GRS",
    "Standard_RAGRS",
    "Standard_ZRS",
    "Premium_LRS",
    "Premium_ZRS",
    "Standard_GZRS",
    "Standard_RAGZRS"
  ]
}
```

Aquí se muestran los valores que este parámetro permitirá. Si la plantilla se ejecuta con un valor no permitido, se producirá un error en la implementación.

4. Agregue un comentario a este parámetro.

```
// This is the allowed values for an Azure storage account
"storageSKU": {
  "type": "string",
  "defaultValue": "Standard_LRS",
  "allowedValues": [
    "Standard_LRS",
    "Standard_GRS",
    "Standard_RAGRS",
    "Standard_ZRS",
    "Premium_LRS",
    "Premium_ZRS",
    "Standard_GZRS",
    "Standard_RAGZRS"
  ]
},
```

Las plantillas de ARM admiten comentarios // y /* */.

5. Actualice **resources** para usar el parámetro storageSKU. Aproveche las ventajas de IntelliSense en Visual Studio Code para facilitar esta tarea.

```
"sku": {
  "name": "[parameters('storageSKU')]"
}
```

El archivo completo tendrá este aspecto:

JSONCopiar

```
{
  "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "storageName": {
      "type": "string",
      "minLength": 3,
      "maxLength": 24,
      "metadata": {
        "description": "The name of the Azure storage resource"
      }
    },
    "storageSKU": {
      "type": "string",
      "defaultValue": "Standard_LRS",
      "allowedValues": [
        "Standard_LRS",
        "Standard_GRS",
```

```

        "Standard_RAGRS",
        "Standard_ZRS",
        "Premium_LRS",
        "Premium_ZRS",
        "Standard_GZRS",
        "Standard_RAGZRS"
    ]
  },
  "functions": [],
  "variables": {},
  "resources": [
    {
      "name": "[parameters('storageName')]",
      "type": "Microsoft.Storage/storageAccounts",
      "apiVersion": "2019-06-01",
      "tags": {
        "displayName": "[parameters('storageName')]"
      },
      "location": "[resourceGroup().location]",
      "kind": "StorageV2",
      "sku": {
        "name": "[parameters('storageSKU')]",
        "tier": "Standard"
      }
    }
  ],
  "outputs": {}
}

```

6. Guarde el archivo.

Implementación de la plantilla de ARM

Aquí, se implementará correctamente mediante el uso de un parámetro `storageSKU` que se encuentra en la lista de permitidos. Después, intentará implementar la plantilla mediante un parámetro `storageSKU` que no esté en la lista de permitidos. Se producirá un error en la segunda implementación, como es de esperar.

1. Ejecute los comandos siguientes para implementar la plantilla. Rellene un nombre único para el parámetro `storageName`. Recuerde que este nombre debe ser único en todo Azure. Puede usar el nombre único que ha creado en la última sección. En ese caso, Azure actualizará el recurso en lugar de crear uno.

```

$today=Get-Date -Format "MM-dd-yyyy"
$deploymentName="addSkuParameter-"+$today
New-AzResourceGroupDeployment `
  -Name $deploymentName `
  -TemplateFile $templateFile `
  -storageName {your-unique-name} `
  -storageSKU Standard_GRS

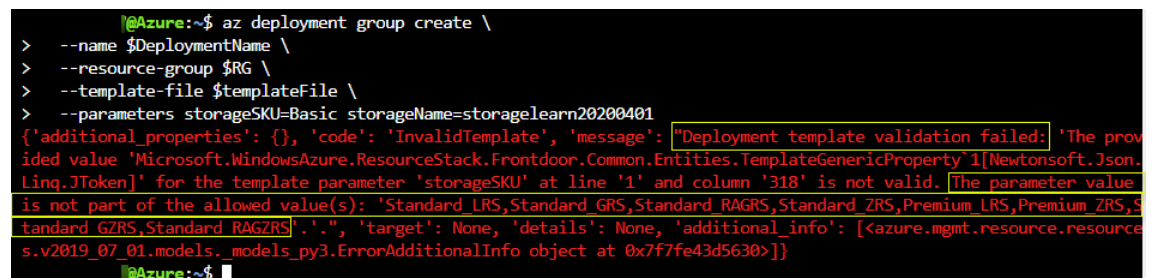
```

Deje que finalice esta implementación. Esta implementación se realiza correctamente de la manera esperada. Los valores permitidos impiden que los usuarios de la plantilla pasen valores de parámetro que no funcionan para el recurso. Ahora se verá lo que sucede cuando se proporciona una SKU no válida.

2. Ejecute los comandos siguientes para implementar la plantilla con un parámetro que no está permitido. Aquí ha cambiado el parámetro `storageSKU` por **Basic**. Rellene un nombre único para el parámetro `storageName`. Recuerde que este nombre debe ser único en todo Azure. Puede usar el nombre único que ha creado en la última sección. En ese caso, Azure actualizará el recurso en lugar de crear uno.

```
$today=Get-Date -Format "MM-dd-yyyy"
$deploymentName="addSkuParameter-"+$today
New-AzResourceGroupDeployment `
  -Name $deploymentName `
  -TemplateFile $templateFile `
  -storageName {your-unique-name} `
  -storageSKU Basic
```

Se produce un error en esta implementación. Observe el error.



```
@Azure:~$ az deployment group create \
> --name $DeploymentName \
> --resource-group $RG \
> --template-file $templateFile \
> --parameters storageSKU=Basic storageName=storagelearn20200401
{'additional_properties': {}, 'code': 'InvalidTemplate', 'message': "Deployment template validation failed: 'The provided value 'Microsoft.WindowsAzure.ResourceStack.Frontdoor.Common.Entities.TemplateGenericProperty 1[Newtonsoft.Json.Linq.JToken]' for the template parameter 'storageSKU' at line '1' and column '318' is not valid. The parameter value is not part of the allowed value(s): 'Standard LRS,Standard GRS,Standard RAGRS,Standard ZRS,Premium LRS,Premium ZRS,Standard GZRS,Standard RAGZRS'.'.", 'target': None, 'details': None, 'additional_info': [<azure.mgmt.resource.resources.v2019_07_01.models._models_py3.ErrorAdditionalInfo object at 0x7f7fe43d5630>]}
```

Adición de salida a la plantilla de ARM

Aquí agregará a la sección `outputs` de la plantilla de ARM para generar los puntos de conexión para el recurso de cuenta de almacenamiento.

1. En el archivo `azuredeploy.json` de Visual Studio Code, coloque el cursor entre las llaves del atributo `outputs` `"outputs":{}`.
2. Presione Entrar y luego escriba `out`. Obtendrá una lista de fragmentos de código relacionados. Seleccione **new-output**. Esto agrega una salida genérica a la plantilla. Tendrá este aspecto:

```
"outputs": {
  "output1": {
    "type": "string",
    "value": "value"
```

```
}
```

3. Cambie **"output1"** a **"storageEndpoint"**, y después cambie el valor de type a **"object"**. Cambie el valor de value a **"[reference(parameters('storageName')).primaryEndpoints]"**. Esta expresión es la que se describe en la unidad anterior que obtiene los datos del punto de conexión. Dado que especificamos *object* como tipo, devolverá el objeto en formato JSON.

```
"outputs": {  
  "storageEndpoint": {  
    "type": "object",  
    "value":  
    "[reference(parameters('storageName')).primaryEndpoints]"  
  }  
}
```

4. Guarde el archivo.

Implementación de la plantilla de ARM con una salida

Aquí, implementará la plantilla y verá la salida de los puntos de conexión como JSON. Tendrá que rellenar un nombre único para el parámetro `storageName`. Recuerde que este nombre debe ser único en todo Azure. Puede usar el nombre único que ha creado en la última sección. En ese caso, Azure actualizará el recurso en lugar de crear uno.

1. Ejecute los comandos siguientes para implementar la plantilla. Asegúrese de reemplazar *{your-unique-name}* (el nombre único) por una cadena única.

```
$today=Get-Date -Format "MM-dd-yyyy"  
$deploymentName="addOutputs-"+$today  
New-AzResourceGroupDeployment `  
  -Name $deploymentName `  
  -TemplateFile $templateFile `  
  -storageName {your-unique-name} `  
  -storageSKU Standard_LRS
```

Observe la salida.

```
],  
  "outputs": {  
    "storageEndpoint": {  
      "type": "Object",  
      "value": {  
        "blob": "https://storelearnv1mlemv4t3u7i.blob.core.windows.net/",  
        "dfs": "https://storelearnv1mlemv4t3u7i.dfs.core.windows.net/",  
        "file": "https://storelearnv1mlemv4t3u7i.file.core.windows.net/",  
        "queue": "https://storelearnv1mlemv4t3u7i.queue.core.windows.net/",  
        "table": "https://storelearnv1mlemv4t3u7i.table.core.windows.net/",  
        "web": "https://storelearnv1mlemv4t3u7i.z22.web.core.windows.net/"  
      }  
    }  
  }  
}
```

Comprobación de la implementación de salida

En Azure Portal, vaya a la implementación *addOutputs*. Ahí también puede encontrar la salida.

Inicio > Grupos de recursos > learn-f798d92d-109a-4072-921e-ce5bf7ceb038 | Implementaciones >

addoutputs-08-Jun-2020 | Salidas

implementación

Buscar (Cmd+/) << storageEndpoint {"dfs":"https://learnstorage4848484.dfs.core.wind...}

- Información general
- Entradas
- Salidas**
- Plantilla