

Demo: Create ARM templates by using the Azure Portal

In this Demo you will learn how to create, edit, and deploy an Azure Resource Manager template by using the Azure portal. This demo shows you how to create an Azure Storage account, but you can use the same process to create other Azure resources.

Generate a template using the portal

Using the Azure portal, you can configure a resource, for example an Azure Storage account. Before you deploy the resource, you can export your configuration into a Resource Manager template. You can save the template and reuse it in the future.

1. Sign in to the Azure portal: <https://portal.azure.com/>.
2. Select **Create a resource > Storage > Storage account**.
3. Enter the following information.
 - **Resource group:** Select **Create new**, and specify a resource group name of your choice.
 - **Name:** Give your storage account a unique name. The storage account name must be unique across all of Azure. If you get an error message saying “The storage account name is already taken”, try using **<your name>storage<Today's date in MMDD>**, for example *mystorage1016*.
 - You can use the default values for the rest of the properties. **Note:** Some of the exported templates require some edits before you can deploy them.
4. Select **Review + create** on the bottom of the screen.
¶ **Note:** Do **not** select **Create** in the next step.
5. Select **Download a template for automation** on the bottom of the screen. The portal shows the generated template:
 - The main pane shows the template. It is a JSON file with six top-level elements - schema, contentVersion, parameters, variables, resources, and output.
 - There are six parameters defined. One of them is called **storageAccountName**. In the next section, you edit the template to use a generated name for the storage account.
 - In the template, one Azure resource is defined. The type is `Microsoft.Storage/storageAccounts`. Note how the resource is defined and the definition structure.
6. Select **Download** from the top of the screen. Open the downloaded zip file, and then save **template.json** to your computer. In the next section, you use a template deployment tool to edit the template.
7. Select the **Parameter** tab to see the values you provided for the parameters. Write down these values, you need them in the next section when you deploy the template.

Edit and deploy the template

The Azure portal can be used to perform some basic template editing by using a portal tool called *Template Deployment*. To edit a more complex template, consider using Visual Studio Code which provides richer edit functionalities.

Azure requires that each Azure service has a unique name. The deployment fails if you enter a storage account name that already exists. To avoid this issue, you can use the template function `uniquestring()` to generate a unique storage account name.

1. In the Azure portal, select **Create a resource**.
2. In **Search the Marketplace**, type **template deployment**, and then press **ENTER**.

3. Select **Template deployment (deploy using custom templates)**.
4. Select **Create**.
5. Select **Build your own template** to open the editor.
6. Select **Load file**, and then select the *template.json* file you downloaded in the last section.
7. Make the following three changes to the template:
 - Remove the **storageAccountName** parameter from the **parameters** element.
 - Add one variable called **storageAccountName** as shown below to the **variables** element. The example below will generate a unique Storage Account name:

```
"storageAccountName": "[concat(uniqueString(subscription().subscriptionId), 'storage')]"
```
 - Update the name element of the **Microsoft.Storage/storageAccounts** resource to use the newly defined variable instead of the parameter:

```
"name": "[variables('storageAccountName')]"
```
8. Select **Save**.
9. In the **BASICS** section of the form that appears select the resource group you created in the last section.
10. In the **SETTINGS** section of the form enter the values from the parameters you wrote down in Step 8 of the previous section. Here is a screenshot of a sample deployment:

Custom deployment

Deploy from a custom template

Customized template

1 resource

Edit template

Edit parameters

Learn more

BASICS

Subscription

<Azure Subscription Name>

Resource group

mystorage1016rg

Create new

Location

Central US

SETTINGS

Location

Central US

Account Type

Standard_LRS

Kind

Storagev2

Access Tier

Hot

Supports Https Traffic Only

true

TERMS AND CONDITIONS

Azure Marketplace Terms

Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

☒ I agree to the terms and conditions stated above

Purchase

11. Accept the terms and conditions and then select **Purchase**.

12. Select the bell icon (notifications) from the top of the screen to see the deployment status. Wait until the deployment is completed.

13. Select **Go to resource group** from the notification pane. You can see the deployment status was successful, and there is only one storage account in the resource group. The storage account name is a unique string generated by the template.

Clean up resources

When the Azure resources are no longer needed, clean up the resources you deployed by deleting the resource group.

<https://www.skillpipe.com/#/reader/urn:uuid:88438492-2a00-5769-bee1-e4c9ebc889fb@2020-12-12T08:30:18Z/content>

3/3