**Exercise - Run a basic Azure Resource Manager template from Azure Cloud Shell**

In this exercise, you use a template that creates a SQL Server logical instance, an Azure App Service plan, and an App Service instance. You deploy this template from Azure Cloud Shell and see that the resources are deployed. Then you delete all the resources in the resource group.

**Open Cloud Shell through the Azure portal**

Here, you open Cloud Shell through the Azure portal so that you can run your Azure Resource Manager template.

You can also install and run the Azure CLI locally .

1. Go to the Azure portal  and sign in.
2. From the menu bar, select **Cloud Shell**. Select the **Bash** experience.

Azure portal page with the Cloud Shell icon on the menu bar.

**Note**

Cloud Shell requires an Azure storage resource to persist any files that you create while working in Cloud Shell. The first time you open Cloud Shell, it prompts you with an offer to create a resource group, storage account, and Azure Files share on your behalf. This is a one-time step and will be automatically attached for all future Cloud Shell sessions.

**Create a working directory**

Here, you create a directory to hold your Resource Manager template.

1. In Cloud Shell, create a directory named *mslearn-resource-manager-template*.

**Bash**

mkdir ~/mslearn-resource-manager-template

1. Move to the *mslearn-resource-manager-template* directory.

**Bash**

cd ~/mslearn-resource-manager-template

**Add the Resource Manager template**

Here, you create a basic Resource Manager template in a file named *template.json*.

1. From the *mslearn-resource-manager-template* directory, open the editor.

**Bash**

code template.json

1. Add this to *template.json* and then save the file.

**JSON**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"serverfarms\_tailspin\_space\_game\_asp\_name": {

"defaultValue": "tailspin-space-game-asp",

"type": "String"

},

"servers\_tailspin\_space\_game\_sql\_name": {

"defaultValue": "tailspin-space-game-sql",

"type": "String"

},

"sites\_tailspin\_space\_game\_web\_name": {

"defaultValue": "tailspin-space-game-web",

"type": "String"

},

"location": {

"type": "string",

"defaultValue": "[resourceGroup().location]"

},

"deployPrefix": {

"type": "string",

"minLength": 3,

"maxLength": 11

},

"uniqueSuffix": {

"type": "string",

"minLength": 3,

"maxLength": 11

},

"adminPassword": {

"type": "string"

}

},

"variables": {

"uniqueName": "[concat(parameters('deployPrefix'), parameters('uniqueSuffix'))]"

},

"resources": [

{

"type": "Microsoft.Sql/servers",

"apiVersion": "2015-05-01-preview",

"name": "[concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName'))]",

"location": "[parameters('location')]",

"kind": "v12.0",

"properties": {

"administratorLogin": "azuresql",

"administratorLoginPassword": "[parameters('adminPassword')]",

"version": "12.0"

}

},

{

"type": "Microsoft.Web/serverfarms",

"apiVersion": "2016-09-01",

"name": "[parameters('serverfarms\_tailspin\_space\_game\_asp\_name')]",

"location": "[parameters('location')]",

"sku": {

"name": "B1",

"tier": "Basic",

"size": "B1",

"family": "B",

"capacity": 1

},

"kind": "app",

"properties": {

"name": "[parameters('serverfarms\_tailspin\_space\_game\_asp\_name')]"

}

},

{

"type": "Microsoft.Sql/servers/firewallRules",

"apiVersion": "2015-05-01-preview",

"name": "[concat(concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName')), '/AllowAllWindowsAzureIps')]",

"dependsOn": [

"[resourceId('Microsoft.Sql/servers', concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName')))]"

],

"properties": {

"startIpAddress": "0.0.0.0",

"endIpAddress": "0.0.0.0"

}

},

{

"type": "Microsoft.Web/sites",

"apiVersion": "2016-08-01",

"name": "[concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName'))]",

"location": "[parameters('location')]",

"dependsOn": [

"[resourceId('Microsoft.Web/serverfarms', parameters('serverfarms\_tailspin\_space\_game\_asp\_name'))]"

],

"kind": "app",

"properties": {

"enabled": true,

"hostNameSslStates": [

{

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '.azurewebsites.net')]",

"sslState": "Disabled",

"hostType": "Standard"

},

{

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '.scm.azurewebsites.net')]",

"sslState": "Disabled",

"hostType": "Repository"

}

],

"serverFarmId": "[resourceId('Microsoft.Web/serverfarms', parameters('serverfarms\_tailspin\_space\_game\_asp\_name'))]"

}

},

{

"type": "Microsoft.Web/sites/hostNameBindings",

"apiVersion": "2016-08-01",

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '/', concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName'), '.azurewebsites.net'))]",

"location": "[parameters('location')]",

"dependsOn": [

"[resourceId('Microsoft.Web/sites', concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')))]"

],

"properties": {

"siteName": "tailspin-space-game-web",

"hostNameType": "Verified"

}

}

]

}

**Understand the template**

Here, you briefly walk through each section of the Resource Manager template to better understand how it works.

The parameters section of the template file specifies the information that needs to be passed in to the template. It has parameters for the names of the resources to be created. The defaults are the base names that will have the deployment prefix and unique suffix added to them.

The deployment prefix, unique suffix, and admin password will be passed in. The location will be set by default as the location of the resource group you're deploying into.

**JSON**

"parameters": {

"serverfarms\_tailspin\_space\_game\_asp\_name": {

"defaultValue": "tailspin-space-game-asp",

"type": "String"

},

"servers\_tailspin\_space\_game\_sql\_name": {

"defaultValue": "tailspin-space-game-sql",

"type": "String"

},

"sites\_tailspin\_space\_game\_web\_name": {

"defaultValue": "tailspin-space-game-web",

"type": "String"

},

"location": {

"type": "string",

"defaultValue": "[resourceGroup().location]"

},

"deployPrefix": {

"type": "string",

"minLength": 3,

"maxLength": 11

},

"uniqueSuffix": {

"type": "string",

"minLength": 3,

"maxLength": 11

},

"adminPassword": {

"type": "string"

}

},

The variables section creates a uniqueName variable that's a combination of the deploymentPrefix parameter and the uniqueSuffix parameter. This variable will be used to create unique names for the resources.

**JSON**

"variables": {

"uniqueName": "[concat(parameters('deployPrefix'), parameters('uniqueSuffix'))]"

},

The rest of the file defines the resources that will be created. Take a moment to read through those.

**SQL Server instance**

Notice the name parameter. Here, you use the default name of the resource that you set up in the parameters section and concatenate the uniqueName variable from the variables section. You get the administrator login password from the adminPassword parameter.

**JSON**

"type": "Microsoft.Sql/servers",

"apiVersion": "2015-05-01-preview",

"name": "[concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName'))]",

"location": "[parameters('location')]",

"kind": "v12.0",

"properties": {

"administratorLogin": "azuresql",

"administratorLoginPassword": "[parameters('adminPassword')]",

"version": "12.0"

}

You also set a firewall rule to allow Azure to access the server. You need this rule to deploy the database to the server. We add database deployment to the template when we deploy in the pipeline later.

**JSON**

"type": "Microsoft.Sql/servers/firewallRules",

"apiVersion": "2015-05-01-preview",

"name": "[concat(concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName')), '/AllowAllWindowsAzureIps')]",

"dependsOn": [

"[resourceId('Microsoft.Sql/servers', concat(parameters('servers\_tailspin\_space\_game\_sql\_name'), variables('uniqueName')))]"

],

"properties": {

"startIpAddress": "0.0.0.0",

"endIpAddress": "0.0.0.0"

}

**App Service plan**

The App Service plan does not need the unique suffix. It uses the name of the plan directly from the parameter's default value.

**JSON**

"type": "Microsoft.Web/serverfarms",

"apiVersion": "2016-09-01",

"name": "[parameters('serverfarms\_tailspin\_space\_game\_asp\_name')]",

"location": "[parameters('location')]",

"sku": {

"name": "B1",

"tier": "Basic",

"size": "B1",

"family": "B",

"capacity": 1

},

"kind": "app",

"properties": {

"name": "[parameters('serverfarms\_tailspin\_space\_game\_asp\_name')]"

}

**App Service instance**

Here, the name property concatenates the name of the website from the parameter default with the unique suffix. You also disable SSL in the hostNameSslStates property.

**JSON**

"type": "Microsoft.Web/sites",

"apiVersion": "2016-08-01",

"name": "[concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName'))]",

"location": "[parameters('location')]",

"dependsOn": [

"[resourceId('Microsoft.Web/serverfarms', parameters('serverfarms\_tailspin\_space\_game\_asp\_name'))]"

],

"kind": "app",

"properties": {

"enabled": true,

"hostNameSslStates": [

{

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '.azurewebsites.net')]",

"sslState": "Disabled",

"hostType": "Standard"

},

{

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '.scm.azurewebsites.net')]",

"sslState": "Disabled",

"hostType": "Repository"

}

],

"serverFarmId": "[resourceId('Microsoft.Web/serverfarms', parameters('serverfarms\_tailspin\_space\_game\_asp\_name'))]"

}

The hostNameBindings type sets the website name and the type.

**JSON**

"type": "Microsoft.Web/sites/hostNameBindings",

"apiVersion": "2016-08-01",

"name": "[concat(concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')), '/', concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName'), '.azurewebsites.net'))]",

"location": "[parameters('location')]",

"dependsOn": [

"[resourceId('Microsoft.Web/sites', concat(parameters('sites\_tailspin\_space\_game\_web\_name'), variables('uniqueName')))]"

],

"properties": {

"siteName": "tailspin-space-game-web",

"hostNameType": "Verified"

}

**Tip**

In Cloud Shell, you can close the editor now if you want. But leave the command window open for the next part.

**Select an Azure region for deployment**

A *region* is one or more Azure datacenters within a specific geographic location. East US, West US, and North Europe are examples of regions. Every Azure resource, including an App Service instance or a SQL database, is assigned a region.

To make the commands easier to run, start by selecting a default region. After you specify the default region, later commands use that region unless you specify a different region.

1. From Cloud Shell, run the following az account list-locations command to list the regions that are available from your Azure subscription.

**Azure CLI**

az account list-locations \

--query "[].{Name: name, DisplayName: displayName}" \

--output table

1. From the **Name** column in the output, choose a region that's close to you. For example, choose **eastasia** or **westus2**.
2. Run the following read command to set a Bash variable that's named AZ\_LOCATION.

**Bash**

read AZ\_LOCATION

At the prompt, enter the region that you chose in the previous step.

This Bash variable makes it easier to run the command that comes next. In practice, this Bash variable isn't required.

1. Print the Bash variable to verify that it was set correctly.

**Bash**

echo $AZ\_LOCATION

You see your region, for example, "northeurope".

1. Set your default location for the resources.

**Azure CLI**

az configure --defaults location=$AZ\_LOCATION

**Make your resource names unique**

The name of the SQL Server instance and the web app must be unique.

For learning purposes, here you generate a random number and assign it to the UNIQUE\_ID Bash variable.

1. From Cloud Shell, generate a random number and assign it to the UNIQUE\_ID variable.

**Bash**

UNIQUE\_ID=$RANDOM

1. Print this UNIQUE\_ID to the console. Write it down for later.

**Bash**

echo $UNIQUE\_ID

1. The output resembles this:

**Output**

24536

**Create a random password**

Here, you generate a random password to use with your SQL database.

There are many ways to generate random passwords. The method you choose depends on your workflow and requirements. This method uses the **openssl** tool to generate 32 random bytes and base64 encode the output. Base64 encoding ensures that the result contains only printable characters.

Run the **openssl** tool to generate a random password.

**Bash**

SQL\_PASSWORD=$(openssl rand -base64 32)

**Run the template**

1. Run the following command to create a resource group for the template to deploy into.

**Azure CLI**

az group create --name tailspin-spacegame-web-rg

1. Run the following command to deploy the template.

**Azure CLI**

az deployment group create \

--name deploytemplate \

--resource-group tailspin-spacegame-web-rg \

--template-file template.json \

--parameters deployPrefix="-dev-" uniqueSuffix=$UNIQUE\_ID adminPassword="$SQL\_PASSWORD"

Wait for the deployment to finish. Notice that we're passing in the parameters that we need. Later you'll move them to a parameters file, pipeline variables, and key vault.

1. Go to the Azure portal and select **Resource groups**, and then select the **tailspin-spacegame-web-rg** resource group. Notice that you have an App Service plan, an App Service instance, and a SQL Server instance. Take note of the suffix on each of these resources. It's made from the deployPrefix variable and your uniqueSuffix variable.

Azure portal page with a ist of resources created by the template run.

1. Run the template a second time.

**Azure CLI**

az deployment group create \

--name deploytemplate \

--resource-group tailspin-spacegame-web-rg \

--template-file template.json \

--parameters deployPrefix="-dev-" uniqueSuffix=$UNIQUE\_ID adminPassword="$SQL\_PASSWORD"

Notice that the operation is idempotent, meaning that it doesn't try to create a resource that's already there. However, if you delete a resource or change some of its properties, the new resource or resource changes will be created.

**Clean up**

Here, you delete everything in your resource group, but not the resource group itself. You'll add more resources to your resource group shortly.

From the Azure portal, delete the resources by selecting them in the resource group and then selecting **Delete** from the menu bar. Do not delete the resource group.

When you're prompted, enter **Yes** to confirm. Verify that the resources are deleted but the resource group remains.

Azure portal page showing deleting resources in the resource group.