

Deploying Services using ARM Templates

Overview

In this challenge you will learn the very basics of using Azure Resource Manager (ARM) Templates to deploy services on Azure. ARM Templating allows you to automate the deployment process enabling you to become more agile and deploy new environments much quicker.

In this challenge you will deploy Service Plans, Web Applications, SQL Servers, and SQL Databases.

Prerequisites

The following are required to complete this lab:

- Virtual Machine
- Azure account (<http://azure.microsoft.com/pricing/free-trial>)

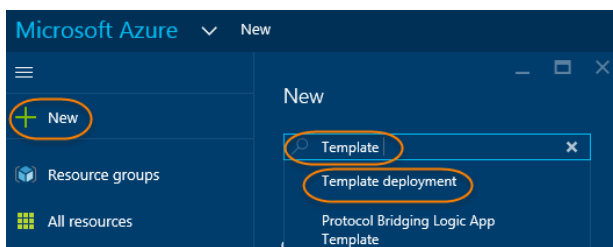
Estimated time to complete this lab: **20** minutes.

Exercise 1: Deploying Services using ARM Templates

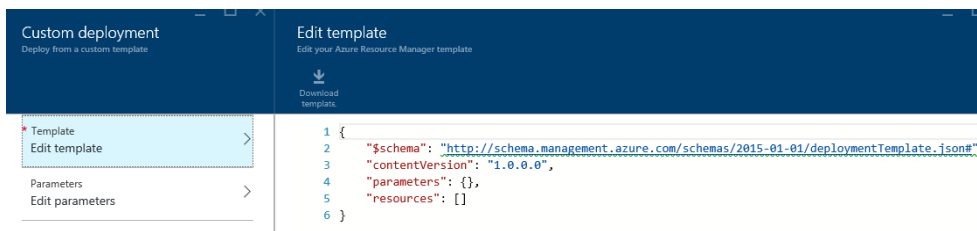
In this task we are going to create a deployment using an ARM Template to deploy a single App Service Plan.

Task 1: Create a single App Service Plan (Hosting Plan)

1. Log in to your Azure account at <https://portal.azure.com>.
2. In the top-left corner of the portal click **+ New**.
3. In the search box at the top, start typing "Template deployment" and select the **Template deployment** option.



4. Click the **Template deployment** item and click **Create**.
5. Click **Edit Template**. A template is a description of all the resources that you want to deploy. When you deploy resources, most of the time you also need to supply parameters. For example, you need to specify the region a resource needs to be deployed into. Templates can also describe dependencies between resources, such as a set of ordered instructions.



6. When you deploy services themselves in Azure, for example a Virtual Machine or a Web App – a template is used under the covers to describe the service and an Azure service called the Azure Resource Manager is used to execute the templates instructions.
7. On the **Edit template** blade locate the line containing this code: "parameters": {}. This should be line 4. Between the braces ("{}") put the following:

```
"hostingPlanName": {
  "type": "string",
  "metadata": {
    "description": "The name of the App Service plan to use for hosting the web
app."
  }
},
"siteLocation": {
  "type": "string",
  "metadata": {
    "description": "The location to use for creating the web app and hosting
plan."
  }
},
"sku": {
  "type": "string",
  "allowedValues": [
    "Free",
    "Shared",
    "Basic",
    "Standard"
  ],
  "defaultValue": "Free",
  "metadata": {
    "description": "The pricing tier for the hosting plan."
  }
},
"workerSize": {
  "type": "string",
  "allowedValues": [
    "0",
    "1",
    "2"
  ],
  "defaultValue": "0",
  "metadata": {
    "description": "The instance size of the hosting plan (small, medium, or
large)."
  }
}
```

8. The configuration above defines the parameters that will be used for the template.
9. Next, update the **resources** property (the area between the brackets "[]") with the configuration below.

```
"resources": [{
  "apiVersion": "2015-04-01",
  "name": "[parameters('hostingPlanName')]",
  "type": "Microsoft.Web/serverFarms",
  "location": "[parameters('siteLocation')]",
  "properties": {
    "name": "[parameters('hostingPlanName')]",
```

```

    "sku": "[parameters('sku')]",
    "workerSize": "[parameters('workerSize')]",
    "numberOfWorkers": 1
  }
}
]

```

10. This configuration defines the services that need to be created, in this case an **Application Plan**.
11. In order to make your life easier for the next tasks, copy the whole configuration into a new instance of **Notepad**.
12. Go back to the portal and press **Save**.
13. From the **Resource group** dropdown, select **+ New**.
14. Enter “templateRG” as the **New resource group location**.
15. Click on **Resource group location** and select any of the available regions.
16. Click the **Edit parameters** link.
17. For the **HOSTINGPLANNAME**, use “templateASP”.
18. For the **SITELOCATION**, use the **same** value you selected for the **Resource group location** which is displayed in the **Resource group location** link (“West US” in this example).

The screenshot shows two side-by-side panels from the Azure portal. The left panel, titled 'Custom deployment', has a sidebar with links for 'Template', 'Parameters', 'Subscription', 'Resource group', and 'Resource group location'. The 'Resource group location' dropdown is set to 'West US'. The right panel, titled 'Parameters', shows the 'HOSTINGPLANNAME' set to 'templateASP', 'SITELOCATION' set to 'West US', 'SKU' set to 'Free', and 'WORKERSIZE' set to '0'. The 'SITELOCATION' dropdown is highlighted with an orange circle.

19. Leave the other values alone and press **OK**
20. Click on **Legal Terms** and press **Create** to indicate that you agree with the Legal Terms.
21. Make sure that **Pin to dashboard** is selected, and then press the **Create**.
22. It will take a moment for your app service plan to become available.

Task 2: Create a single App Service Plan and a Web App

1. You are going to repeat creating the template you did in the previous task, but you have all this copied in **Notepad** ready to go.
2. Create a new template (**+New**, search for “Template deployment”, all the way through the first **Create** to start creating the template).
3. Click **Edit template** and delete everything that is there.

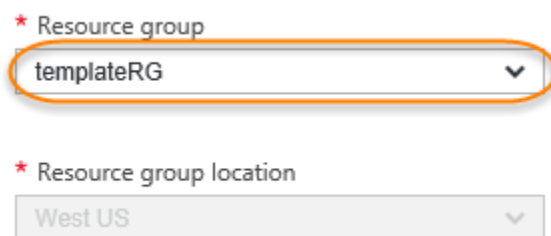
4. Switch to **Notepad** where your old template definition is waiting. Copy it to the portal to replace the existing default template.
5. In the **parameters** section, add the following parameter at the beginning of the list, right after the opening "{".

```
"siteName": {  
  "type": "string",  
  "metadata": {  
    "description": "The name of the web app that you wish to create."  
  }  
},
```

6. In the **resources** section add a comma after the existing resource, directly **before** the last "]" closing square bracket and then add the resource below.

```
{  
  "apiVersion": "2015-06-01",  
  "name": "[parameters('siteName')]",  
  "type": "Microsoft.Web/Sites",  
  "location": "[parameters('siteLocation')]",  
  "dependsOn": [  
    "[concat('Microsoft.Web/serverFarms/', parameters('hostingPlanName'))]"  
  ],  
  "properties": {  
    "name": "[parameters('siteName')]",  
    "serverFarmId": "[parameters('hostingPlanName')]"  
  }  
}
```

7. Save the template.
8. Click on **Resource group** and choose "templateRG", which will fix the location.

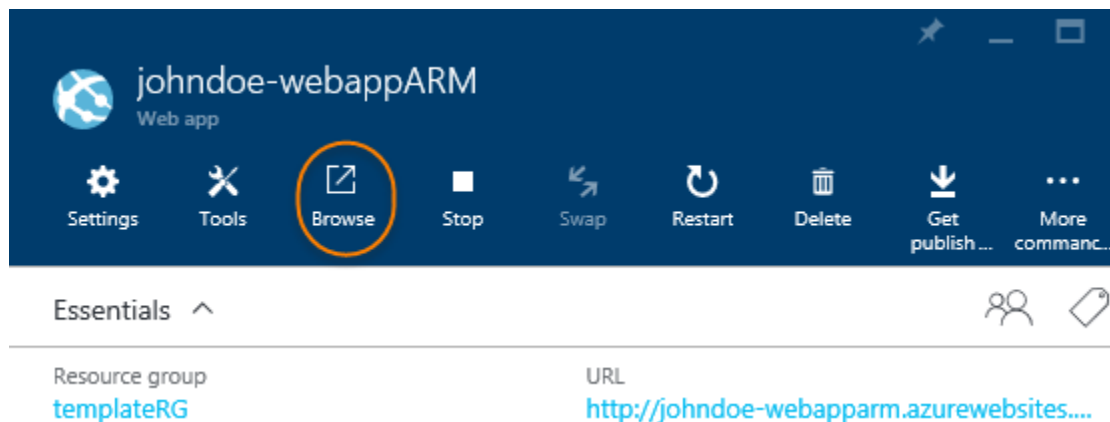


The image shows two dropdown menus from the Azure portal. The first dropdown is labeled "Resource group" and has "templateRG" selected. The second dropdown is labeled "Resource group location" and has "West US" selected. Both dropdowns have a downward arrow icon on the right side.

9. Click the **Edit parameters** link.
10. For **SITENAME**, enter a globally unique name. Consider using your name plus "-webappARM", such as "johndoe-webappARM".
11. For the **HOSTINGPLANNAME**, use "templateASP" – which will fix the location.
12. For the **SITELOCATION** use the **same** value you selected for the Resource Group Location which is displayed in the **Resource group location** link.
13. Leave the other values alone and press **OK**.
14. Click on **Legal Terms** and press **Create** to indicate that you agree with the Legal Terms.
15. Make sure that **Pin to dashboard** is selected, and then press the **Create**.
It will take a moment for your app service plan to become available.

Task 3: Review what you have created

1. Click the **Resource groups** item from the portal's left menu.
2. Click your "templateRG" item to open it.
3. In the **Summary** section **App Service Plan** and your Web app. These were the resources that were described by the template. They were created using the values you supplied to the template based on the parameters the template needed.
4. Click your Web site and click the **Browse** button on the toolbar. Your site opens. It all just works.



5. What you just created is a template that's similar to the template used (under the covers) if you had created it from **+New | Web + Mobile | Web App**.

Task 4: Create a single App Service Plan, a Web App, a SQL Server, and a SQL Database

1. Create a new template (**+New**, search for "Template deployment", all the way through the first **Create** to start creating the template).
2. Click **Edit Template** and delete everything that is there.
3. In a new browser window go to <https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/201-web-app-sql-database/azuredeploy.json>.
4. Copy the whole content from there and paste it in the **Edit template** blade.
5. **Save** the template and go to the **Edit parameters** blade.
6. For the **SITENAME**, use a globally unique name.
7. For **HOSTINGPLANNAME** we suggest you use "azappplan3".
8. For **SITELOCATION** use any one of these 4 values: "West US", "South Central US", "Central US" and "East US".
9. For **SERVERNAME**, use a globally unique name.
10. For **SERVERLOCATION**, use the same location as you used for Site Location.
11. For **ADMINISTRATORLOGIN**, use name + "admin", such as "johndoeadmin". Select a password you can remember.
12. For **DATABASENAME**, we suggest you use name + "-sqldbARM", such as "johndoe-sqldbARM".
13. Press **OK**.
14. From the **Resource group** dropdown, select **+ New**.
15. Enter "FullArmRG" as the **New resource group location**.

16. Click on **Resource group location** and select the same region used in earlier steps.
17. Click on **Legal Terms** and press **Create** to indicate that you agree with the Legal Terms.
18. Make sure that **Pin to dashboard** is selected, and then press the **Create**.
19. After a while, depending on the load on Azure at the time, your deployment is created and a blade with your resource group will be open.