# Lab Answer Key:  Module 6: Resource Manager

## Lab 1: Automating Workloads with a Template

In this lab you will learn how to work with Azure Resource Manager templates. Rather than deploying resources in Azure independently, you can leverage Azure ARM templates to deploy and provision multiple resources for an Azure-based solution in a single, coordinated operation. In addition, you can continue re-deploying templates to either revert inadvertent changes to your solution or to expand the original implementation by adding other resources to it. The ARM template consists of JSON and expressions which you can use to construct values for your deployment.

Note: If you want to create and modify ARM templates, then you should consider using for this purpose development tools, such as Visual Studio. This considerably helps with the development process. However, for the sake of simplicity, we will use the capabilities built into the Azure portal to illustrate the format and some of the core functionality of ARM templates.

Objectives:

* Use ARM templates to deploy, configure, and recreate Azure Resource Manager resources.
* Understand the syntax of ARM templates
* Deploy templates by using the Azure portal

Estimated time for completion: 40 minutes

### Exercise 1: Deploy and Configure Network Resources by using ARM Templates

Your need to create and configure a virtual network by using ARM templates.

The main tasks for this exercise are as follows:

1. Deploy a virtual network by using an ARM template
2. Add a subnet to a virtual network by using an ARM template

#### Task 1: Deploy a virtual network by using an ARM template

1. Launch your Internet browser and navigate to https://portal.azure.com
2. Sign in to the Azure portal by using your administrative credentials.
3. Launch another instance of the Internet browser and navigate to https://github.com/Azure/azure-quickstart-templates/tree/master/101-vnet-two-subnets
4. On the **Virtual Network with two Subnets** page, click **Deploy to Azure**
5. If prompted to sign-in to Microsoft Azure, specify the same credentials you used in step 2.
6. On the **Create a Virtual Network with two Subnets** blade, click Edit
7. On the **Edit template** blade, review the JSON layout section on the left hand side and the actual JSON template content on the right hand side. Note that you can make changes to the template from within this interface and save them in order to modify your deployment. We will not be making changes to the template.
8. Click **Discard**
9. Back in the **Create a Virtual Network with two Subnets** blade, specify the following settings and click **Purchase**:

Note: There is no costs associated with using ARM templates.

* Subscription: the name of the subscription you are using for this lab
* Resource Group: click **Create new** and type **ARMTemplatesVNetRG** in the text box below
* Location: Choose the Azure region closest to your location.
* Vnet Name: **VNet1**
* Vnet Address Prefix: **10.0.0.0/16**
* Subnet1Prefix: **10.0.0.0/24**
* Subnet1Name: **Subnet1**
* Subnet2Prefix: **10.0.1.0/24**
* Subnet2Name: **Subnet2**
* I agree to the terms and conditions stated above: enabled

#### Task 2: Deploy a virtual network by using an ARM template

1. Once the deployment completes, launch another instance of the Internet browser and navigate to https://github.com/Azure/azure-quickstart-templates/tree/master/101-subnet-add-vnet-existing
2. On the **Add a subnet to an existing VNET** page, click **Deploy to Azure**
3. If prompted to sign-in to Microsoft Azure, specify the same credentials you used above.
4. On the **Add a subnet to an existing VNET** blade, click **Edit**:
5. On the **Edit template** blade, review the JSON layout section on the left hand side and the actual JSON template content on the right hand side. Note that, just as before, you can make changes to the template from within this interface and save them in order to modify your deployment.
6. Click **Discard**
7. On the Add a subnet to an existing VNET blade, specify the following settings and click **Purchase**:

* Subscription: the name of the subscription you are using for this lab
* Resource Group: click **Use existing** and select **ARMTemplatesVNetRG** in the text box below
* Location: this will be automatically set once you select the existing resource group.
* Existing VNET Name: **VNet1**
* New Subnet Name: **Subnet3**
* New Subnet Address Prefix: **10.0.2.0/24**
* I agree to the terms and conditions stated above: enabled

1. Once the deployment completes, in the Azure portal, in the hub menu, click **More services**
2. In the list of services, in the **NETWORKING** section, click **Virtual networks**
3. On the **Virtual networks** blade, click **VNet1** in the **ARMTemplatesVNetRG** resource group.
4. On the **VNet1** blade, click **Subnets**
5. Verify that VNet1 consists of three subnets.

Note: It is important to realize that we can apply incremental changes by using pre-created templates because this is how these specific templates were designed. In general, by default, applying a template to an already existing deployment will reset it to the state that is defined in the template. This can be beneficial if you want to restore resources in the deployment to their original state.

Results: In this exercise, you have created and configured a new virtual network with three different subnets by using two different ARM templates.

### Exercise 2: Deploy and configure compute resources by using ARM templates

In this exercise you will continue the work you started in the previous exercise by deploying a virtual machine into an existing virtual network

The main tasks for this exercise are as follows:

1. Deploy a virtual machine in to an existing virtual network by using an ARM template
2. Save the deployment template representing the VM deployment
3. Use the template to recreate a missing resource

#### Task 1: Deploy a virtual machine in to an existing virtual network by using an ARM template

1. Launch another instance of the Internet browser and navigate to https://github.com/Azure/azure-quickstart-templates/tree/master/201-vm-different-rg-vnet
2. On the **Create a VM in a VNET in a different Resource Group** page, click **Deploy to Azure**
3. If prompted to sign-in to Microsoft Azure, specify the same credentials you used above.
4. On the **Create a VM in a VNET in a different Resource Group** blade, click **Edit**
5. On the **Edit template** blade, review the JSON layout section on the left hand side and the actual JSON template content on the right hand side. Note that you can make changes to the template from within this interface and save them in order to modify your deployment. We will not be making changes to the template at this point.
6. Click **Discard**
7. Back on the **Create a VM in a VNET in a different Resource Group** blade, specify the following settings and click **Purchase**:

* Subscription: the name of the subscription you are using for this lab
* Resource Group: click **Create new** and type **ARMTemplatesVMRG** in the text box below
* Location: Choose the same location you used in the first exercise of this lab.
* New Storage Account Name: a unique string between 3 and 24 characters in length consisting of lower case letters and digits
* Storage Account Type: **Standard\_LRS**
* Public IP Name: **VM1PIP**
* Public IP Address Type: **Dynamic**
* Vm Name: **VM1**
* Vm Size: **Standard\_A0**
* Image Publisher: **MicrosoftWindowsServer**
* Image Offer: **WindowsServer**
* Image SKU: **2012-R2-Datacenter**
* Admin User Name: **demouser**
* Admin Password: **DemoPa$$w0rd**
* Virtual Network Name: **VNet1**
* Virtual Network Resource Group: **ARMTemplatesVNetRG**
* Subnet1Name: **Subnet1**
* Nic Name: **VM1NIC**
* I agree to the terms and conditions stated above: enabled

1. In the Azure portal, in the hub menu, click **Resource groups**
2. On the **Resource groups** blade, click **ARMTemplatesVMRG**
3. Wait for the deployment of the VM to complete.

#### Task 2: Save the deployment template representing the VM deployment

1. In the Azure portal, on the **ARMTemplatesVMRG** blade, click **Automation script**
2. On the **Automation script** blade, click **Add to library**

Note: You can also download the template directly from the Azure portal.

1. On the **Save template** blade, specify the following and click **Save**:

* Name: **VM1\_Deployment**
* Description: **Lab VM1 deployment**

#### Task 3: Use the template to recreate a missing resource

1. In the Azure portal, back on the **ARMTemplatesVMRG** blade, click **Overview** and then, in the list of resources, click **VM1PIP**
2. On the **VM1PIP** blade, click **Dissociate**
3. When prompted to confirm, click **Yes**. Wait until the change takes effect.
4. On the **VM1PIP** blade, click **Delete**
5. When prompted to confirm, click **Yes**

Note that the resource group no longer contains the public IP address resource.

1. In the Azure portal, in the hub menu, click **More services**
2. In the list of services, in the **OTHER** section, click **Templates**
3. On the **Templates** blade, click **vm1\_deployment**
4. On the **vm1\_deployment** blade, click **Deploy**
5. In the **Resource group** section, click **Use existing**, and, in the drop down list, click **ARMTemplatesVMRG**
6. In the **Virtual Machines\_VM1\_admin Password** text box, type **DemoPa$$w0rd**
7. In the **Storage Accounts\_yourstorageaccount\_name** text box, type a unique name of the storage account (verify that there is a green checkmark next to the account name)

Note that you are being prompted to provide a new storage account name due to the way the Azure portal validates the template. You could avoid this by deploying the template via Azure PowerShell or Azure CLI.

1. Enable the **I agree to the terms and conditions stated above** check box.
2. Click **Purchase**
3. In the Azure portal, in the hub menu, click **Resource groups**
4. On the **Resource groups** blade, click **ARMTemplatesVMRG**
5. Wait for the deployment of the VM to complete.
6. Note that the resource group includes the public IP address resource.

Note: Disregard the error message *Changing property 'osDisk.vhd.uri' is not allowed*. This is the result of the limitation of the Azure portal we mentioned earlier which requires providing a name of a new storage account. Despite the error and the failed deployment, the public IP address is successfully created and associated with the network interface of the Azure VM. As we pointed out earlier, you can avoid the error message by using Azure PowerShell or Azure CLI to deploy the template.

Results: In this lab you deployed a virtual machine into an existing virtual network by using an ARM template. You also generated a template representing an existing deployment and used it to recreate a missing resource.

Leave the virtual machine VM1 running. You will need it in the next lab of this module.