**Exercise - Analyze a scenario and map it to a technology option**

Completed600 XP

* 10 minutes

Sandbox activated! Time remaining:

16 min

You have used 1 of 10 sandboxes for today. More sandboxes will be available tomorrow.

Your organization grows, and demand rises in Europe and Asia. Your organization is based in the United States. Most of your infrastructure is cloud based. Your European users are reporting a slow website. To provide better service to the customers in Europe, your company has asked you to create a new and improved web server for Europe. The machine must be configured as an IIS web server when it's provisioned.

You've been asked to choose appropriate provisioning tools. Your tools should also be easy to use and easily customizable.

In this exercise, you'll provision a web server to meet the requirements.

**Provision a web server**

You'll use a couple of tools together. With Azure Resource Manager templates, you can create a template that outlines the environment for your web server. Your Resource Manager template can also help you define a state that you apply to your web server at the point of provisioning. Apply your desired state by linking to a Desired State Configuration (DSC) extension handler inside your Resource Manager template. The DSC handler helps you enforce a state that you define in a DSC configuration.

In the following exercise, you use a Resource Manager template to provision a virtual machine. The DSC extension handler that's included in the template enforces your state on the virtual machine. You use an Azure storage account to host your DSC configuration file.

**Clone the configuration and template**

Let's start by obtaining a configuration file and Resource Manager template from GitHub.

1. In the Azure Cloud Shell instance on the right, run this command to clone the GitHub repository.

BashCopy

git clone https://github.com/MicrosoftDocs/mslearn-choose-compute-provisioning.git

**Examine and compress the state configuration file**

The state configuration is defined in the *Webserver.ps1* file, which includes the IIS web server on your virtual machine. Let's examine that file.

1. To open the file, run these commands:

BashCopy

cd mslearn-choose-compute-provisioning

code Webserver.ps1

1. The configuration file includes the following code. Notice that it installs IIS, ASP.NET, and the IIS Management Console.

PowerShellCopy

Configuration Webserver

{

param ($MachineName)

Node $MachineName

{

#Install the IIS Role

WindowsFeature IIS

{

Ensure = "Present"

Name = "Web-Server"

}

#Install ASP.NET 4.5

WindowsFeature ASP

{

Ensure = "Present"

Name = "Web-Asp-Net45"

}

WindowsFeature WebServerManagementConsole

{

Name = "Web-Mgmt-Console"

Ensure = "Present"

}

}

}

1. To close the code editor, select Ctrl+q.
2. To see the contents of the cloned repository, enter this command:

BashCopy

ls -al

Notice that the repository contains both the *Webserver.ps1* file, and the *Webserver.zip* compressed file. You use the zipped file later in this exercise.

**Configure the template**

The GitHub repository that you cloned also contains a Resource Manager template. Let's add the location of the zipped configuration file to it.

1. In Cloud Shell, use this command to open the template:

BashCopy

code template.json

1. In the *template.json* code file, in the <parameters> section, notice that some properties, such as vmName, don't include default values. When you use the template, you're prompted to enter values for those properties.
2. To provide a default value for the configuration file URL, in the *template.json* code file, locate this code:

JSONCopy

"modulesUrl": {

"type": "string",

"metadata": {

"description": "URL for the DSC configuration module."

}

},

1. Replace that code with this code:

JSONCopy

"modulesUrl": {

"type": "string",

"defaultValue": "https://github.com/MicrosoftDocs/mslearn-choose-compute-provisioning/raw/master/Webserver.zip",

"metadata": {

"description": "URL for the DSC configuration module."

}

},

1. To save the template, select Ctrl+s.
2. In the <variables> section of the *template.json* code file, notice that values are stored to configure networking for the virtual machine.
3. In the <resources> section, notice that the template creates an IP address, a virtual network, the virtual machine, and other related resources.
4. To close the code editor, select Ctrl+q.

**Validate your template**

Now that you have a completed template and zipped configuration file, you can use the Azure CLI to perform a deployment. Before you deploy, validate your files to see if the deployment will be successful.

1. In Cloud Shell, run the following command to validate your deployment. When prompted for a password, enter a complex password of your choice.

Azure CLICopy

az deployment group validate \

--resource-group learn-3accfc27-e031-49b7-a913-86077a172111 \

--template-file template.json \

--parameters vmName=hostVM1 adminUsername=serveradmin

1. If your deployment is validated, you'll see information about your deployment. Pay special attention to the error property, which can be found by scrolling back through the output text. It should be null.

Screenshot of Cloud Shell showing a successful template validation with a callout highlighting the error property set to null.

1. If there are no errors, your template was validated and can be deployed.

**Deploy your template**

Now that we know the template is valid, we can perform the deployment.

1. To deploy the template, run the following command. When prompted for a password, enter a complex password of your choice.

Azure CLICopy

az deployment group create \

--resource-group learn-3accfc27-e031-49b7-a913-86077a172111 \

--template-file template.json \

--parameters vmName=hostVM1 adminUsername=serveradmin

The template takes several minutes to deploy.

1. When the deployment is finished, you'll have a virtual machine configured as an IIS web server. To list all of the resources in the resource group and confirm that everything has been set up, run this command:

Azure CLICopy

az resource list \

--resource-group learn-3accfc27-e031-49b7-a913-86077a172111 \

--output table \

--query "[\*].{Name:name, Type:type}"

You'll see all of your resources listed, which means your deployment was successful.

1. Run this command to generate the URL for your web server so that you can confirm IIS was successfully installed:

Azure CLICopy

echo http://$(az vm show \

--show-details \

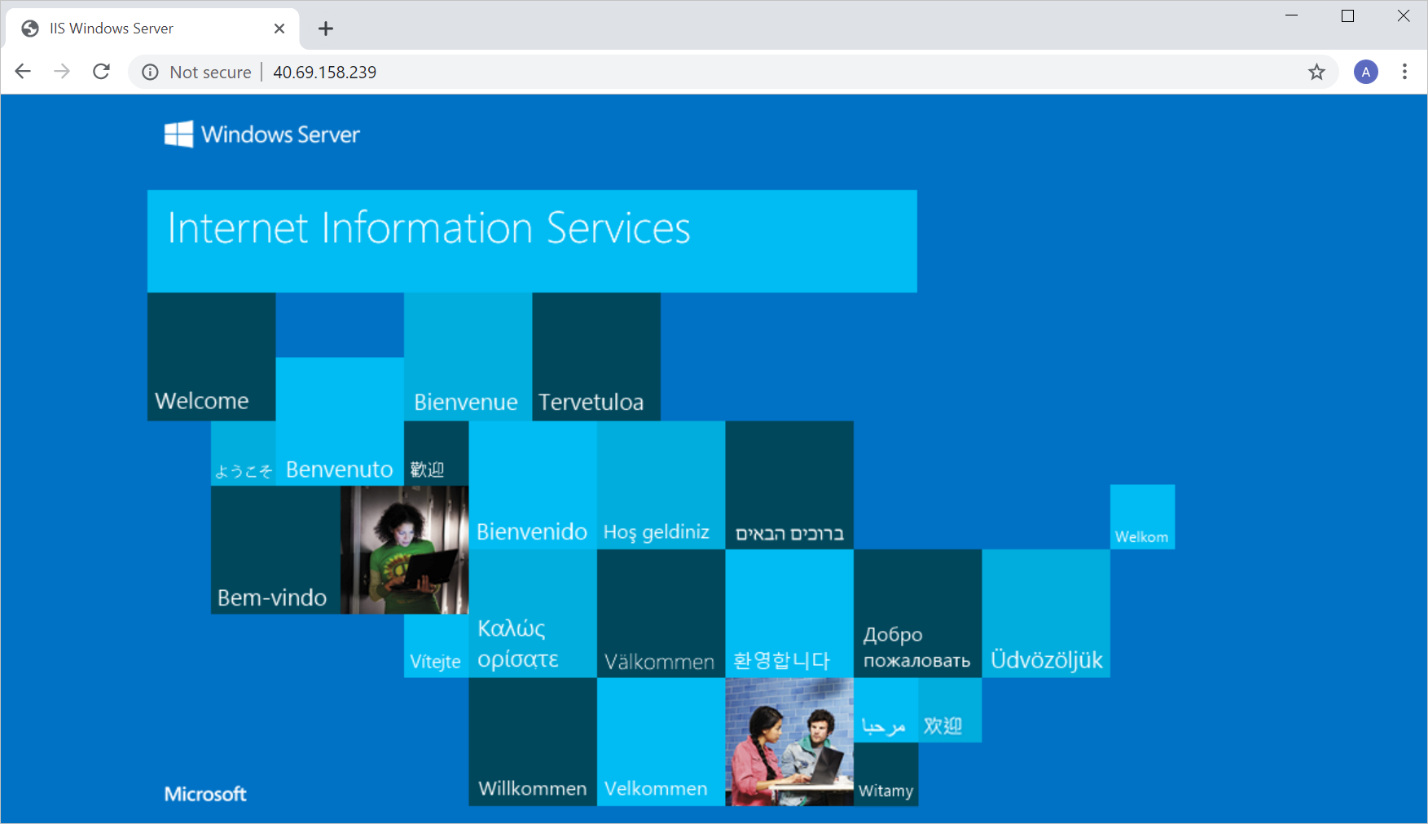
--resource-group learn-3accfc27-e031-49b7-a913-86077a172111 \

--name hostVM1 \

--query publicIps \

--output tsv)

1. Select the URL, or copy and paste it into a new browser window. You should see the default IIS page.



You provisioned a web server by using Resource Manager templates and enforced a desired state configuration on your machine through a DSC extension handler.