AZ-101.2 - Implement and Manage Application Services

Lab: Implement and Manage Azure Web Apps

Estimated Time: 90 minutes

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

Note: If you are not using the Cloud Shell you will need to install the Azure PowerShell module - https://docs.microsoft.com/en-us/powershell/azure/install-azurermps-ps?view=azurermps-6.12.0. You will also need to authenticate to Azure.

Note: Although not required, it is a good idea to deprovision any existing resources when you have completed the lab. This will help mitigate the risk of exceeding the default vCPU quota limits.

Lab files: none

Scenario

Adatum Corporation wants to implement Azure web apps and configure them for scalability and performance.

Objectives

After completing this lab, you will be able to:

- Implement Azure web apps
- Manage scalability and performance of Azure web apps

Exercise 1: Implement Azure web apps

The main tasks for this exercise are as follows:

1. Create an Azure web app
2. Configure web app deployment settings.
3. Create a staging deployment slot
4. Deploy code to the staging deployment slot and perform slot swap

Task 1: Create an Azure web app

Note: You will need to use a Microsoft account that has the Owner role in the Azure subscription you intend to use in this lab and is a Global Administrator of the Azure AD tenant associated with that subscription.

2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Web app**.
4. Use the list of search results to navigate to the **Web App Create** blade.
5. From the **Web App Create** blade, create a new web app with the following settings:
   - App name: any unique, valid name
   - Subscription: the name of the Azure subscription you intend to use in this lab
   - Resource group: the name of a new resource group `az1010201-RG`
   - OS: **Windows**
   - Publish: **Code**
   - App Service plan/Location: a new App Service plan with the following settings:
     - Name: `az1010201-AppServicePlan1`
     - Location: the name of an Azure region where you can provision Azure web apps
     - Pricing tier: **S1 Standard**
   - Application Insights: **Disabled**

Note: The green check mark in the **App name** text box will indicate whether the name you typed in is valid and unique.

Note: To identify Azure regions where you can provision Azure web apps, refer to [https://azure.microsoft.com/en-us/regions/offers/](https://azure.microsoft.com/en-us/regions/offers/)

Note: Wait until the web app is created before you proceed to the next task. This should take about a minute.

### Task 2: Create a staging deployment slot

1. In the Azure portal, navigate to the blade of the newly provisioned web app.
2. On the web app blade, use the **Browse** toolbar icon to open a new browser tab displaying the default App Service home page.
3. Close the browser tab displaying the default App Service home page.
4. In the Azure portal, from the web app blade, display its **Deployment slots (preview)** blade.
5. From the **Deployment slots (Preview)** blade, add a slot with the following settings:
   - Name: **staging**
Clone settings from: **Do not clone settings**

6. From the **Deployment slots** blade, navigate to the **staging** blade displaying the properties of the newly created deployment slot.
7. On the **staging** blade, use the **Browse** toolbar icon to open a new browser tab displaying the default App Service home page.
8. Close the browser tab displaying the default App Service home page in the staging deployment slot.

**Task 3: Configure web app deployment settings.**

1. In the Azure portal, from the **staging** blade, display the **staging - Deployment Center** blade.
2. On the **staging - Deployment Center** blade, configure the following settings:
   - *Source control:* **Local Git**
   - *Build provider:* **App Service Kudu build server**
3. From the **Deployment Center** blade, use the **Deployment Credentials** toolbar icon to display the **User Credentials** pane.
4. In the **User Credentials** pane, specify a unique username and set the password to `Pa55w.rd1234` and save the newly set credentials.
5. Back on the **Deployment Center** blade, note the **Git Clone Url**. You will need in the next task of this exercise.

**Task 4: Deploy code to the staging deployment slot and perform a slot swap**

1. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

   **Note:** If this is the first time you are launching the Cloud Shell in the current Azure subscription, you will be asked to create an Azure file share to persist Cloud Shell files. If so, accept the defaults, which will result in creation of a storage account in an automatically generated resource group.

2. In the Cloud Shell pane, run the following command:

   ```
   git clone https://github.com/Azure-Samples/php-docs-hello-world
   ```

   **Note:** This command clones a remote repository containing a sample web app code

3. In the Cloud Shell pane, run the following command:

   ```
   Set-Location -Path $HOME/php-docs-hello-world/
   ```

   **Note:** This command sets the current location to the newly created clone of the local repository containing the sample web app code
4. In the Cloud Shell pane, run the following command, replacing the `<git_clone_url>` placeholder with the value of the Git Clone Url you identified in the previous task:

```
git remote add azure <git_clone_url>
```

**Note:** This command connects the local repo to the Git repo in Azure repos associated with the Azure web app.

5. In the Cloud Shell pane, run the following command:

```
git push azure master
```

**Note:** This command pushes the sample web app code from the local repository to the Azure web app staging deployment slot.

6. When prompted, type the password `Pa55w.rd1234` you set in the previous task.

7. Wait for the push to complete and close the Cloud Shell pane.

8. In the Azure portal, navigate to the Overview section of the staging blade.

9. On the staging blade, use the Browse toolbar icon to open a new browser tab. Note that the browser displays the Hello World! page you just deployed.

10. Close the browser tab displaying the Hello World! page.

11. On the staging blade, use the Swap toolbar icon to display the Swap blade.

12. On the Swap blade, initiate slot swap with the following settings:
   - Swap type: Swap
   - Source: staging
   - Destination: production
   - Preview Changes: review the changes that will be applied to the destination slot

13. Wait for swap to complete.

14. On the staging blade, use the Browse toolbar icon to open a new browser tab. Note that now the browser displays the default App Service home page.

15. Close the browser tab displaying the default App Service home page.

16. From the staging blade, display its Deployment slots (Preview) blade.

17. From the Deployment slots (Preview) blade, navigate back to the blade displaying the properties of the production slot of the Azure web app.

18. On the web app blade, use the Browse toolbar icon to open a new browser tab. Note that the browser displays the Hello World! page.

19. Close the browser tab displaying the Hello World! page.

**Result:** After you completed this exercise, you have created an Azure web app, configured web app deployment settings, created a staging deployment slot, deployed code to the staging deployment slot, and performed a slot swap.
Exercise 2: Manage scalability and performance of Azure web apps

The main tasks for this exercise are as follows:

1. Configure and test autoscaling of the Azure web app
2. Configure Content Delivery Network (CDN) for the Azure web app

Task 1: Configure and test autoscaling of the Azure web app

1. In the Azure portal, from the web app blade, display the Scale out (App Service plan) blade.
2. On the web app Scale out (App Service plan) blade, enable autoscale with the following settings:
   - Autoscale setting name: az1010201-AutoScaling
   - Resource group: az1010201-RG
   - Default Auto created scale condition:
     - Scale mode: Scale based on a metric
     - Rules: Scale out
     - Criteria:
       - Time aggregation: Total
       - Metric name: Data In
       - Time grain statistic: Maximum
       - Operator: Greater than
       - Threshold: 1024 bytes
       - Duration: 5 minutes
   - Action:
     - Operation: Increase count by
     - Instance count: 1
     - Cool down (minutes): 5
   - Instance limits (Minimum): 1
   - Instance limits (Maximum): 2
   - Instance limits (Default): 1

Note: These settings are chosen to ensure that autoscaling is triggered as soon as possible, rather than to represent optimal settings in real-life scenarios. For guidance regarding the best autoscaling practices, refer to https://docs.microsoft.com/en-us/azure/architecture/best-practices/auto-scaling

3. From the Azure Portal, start a PowerShell session in the Cloud Shell.
4. In the Cloud Shell pane, run the following commands:

```bash
$resourceGroup = Get-AzureRmResourceGroup -Name 'az1010201-RG'
$webapp = Get-AzureRmWebApp -ResourceGroupName $resourceGroup.ResourceGroupName
while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }
```
Note: These commands submit requests to the Azure web app in a loop in order to trigger autoscaling.

5. Minimize the Cloud Shell pane and, from the web app blade, display the Process explorer blade. This will allow you to monitor the number of instances and their resource utilization.
6. Once you notice that the number of instances has increased to 2, reopen the Cloud Shell pane and terminate the PowerShell script by pressing CTRL+C.

Note: It might take about 5 minutes for the number of instances to increase to 2.

Task 2: Configure Content Delivery Network (CDN) for the Azure web app

1. In the Azure portal, navigate to the New blade.
2. From the New blade, search Azure Marketplace for CDN.
3. Use the list of search results to navigate to the CDN profile blade.
4. From the CDN profile blade, create a CDN profile with the following settings:
   o App name: az1010202cdn-profile
   o Subscription: the name of the Azure subscription you are using in this lab
   o Resource group: the name of a new resource group az1010202-RG
   o Location: the name of an Azure region where you can provision Azure CDN
   o Pricing tier: Standard Microsoft
   o Create a new CDN endpoint now: enabled
   o CDN endpoint name: any unique, valid name consisting of letters and digits
   o Origin type: Web App
   o Origin hostname: the fully qualified name of the web app you created in the first exercise

Note: The green check mark in the CDN endpoint name text box will indicate whether the name you typed in is valid and unique.

Note: Wait until the CDN endpoint is created before you proceed to the next task. This should take less than a minute.

5. In the Azure portal, navigate to the az1010202cdn-profile blade.
6. From the az1010202cdn-profile blade, note the list of endpoints and use it to navigate to the newly created endpoint.
7. From the endpoint blade, note the value of the Endpoint hostname.
8. Open a new tab of Microsoft Edge and browse to the URL representing the Endpoint hostname. Note that the browser displays the Hello World! page you just deployed.
**Note:** You might need to refresh the page a few times.

**Result:** After you completed this exercise, you have configured and tested autoscaling of the Azure web app as well as configured Content Delivery Network (CDN) for the Azure web app.