Implementing and Managing Application Services

Lab: Implementing Azure Logic Apps

Scenario

Adatum Corporation wants to implement custom monitoring of changes to a resource group.

Objectives

After completing this lab, you will be able to:

- Create an Azure logic app
- Configure integration of an Azure logic app and an Azure event grid

Lab Setup

Estimated Time: 45 minutes

User Name: **Student**

Password: **Pa55w.rd**

Exercise 1: Set up the lab environment that consists of an Azure storage account and an Azure logic app

The main tasks for this exercise are as follows:

1. Create an Azure storage account
2. Create an Azure logic app
3. Create an Azure AD service principal
4. Assign the Reader role to the Azure AD service principal
5. Register the Microsoft.EventGrid resource provider

Task 1: Create a storage account in Azure

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. From Azure Portal, create a new storage account with the following settings:
   - Subscription: the name of the target Azure subscription
   - Resource group: a new resource group named **az3000701-LabRG**
   - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits

- Location: the name of the Azure region that is available in your subscription and which is closest to the lab location
- Performance: **Standard**
- Account kind: **Storage (general purpose v1)**
- Replication: **Locally-redundant storage (LRS)**
- Secure transfer required: **Enabled**
- Virtual network: **None**
- Hierarchical namespace: **Disabled**

Note: Do not wait for the deployment to complete but instead proceed to the next task.

Task 2: Create an Azure logic app

1. From Azure Portal, create an instance of **Logic App** with the following settings:

   - Name: **logicapp3000701**
   - Subscription: the name of the target Azure subscription
   - Resource group: the name of a new resource group **az3000702-LabRG**
   - Location: the same Azure region into which you deployed the storage account in the previous task
   - Log Analytics: **Off**

2. Wait until the vault is provisioned. This will take about a minute.

Task 3: Create an Azure AD service principal

1. In the Azure portal, in the Microsoft Edge window, start a **PowerShell** session within the **Cloud Shell**.

2. If you are presented with the **You have no storage mounted** message, configure storage by clicking on **Show advanced settings** and using the following settings:

   - Subsciption: the name of the target Azure subscription
   - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location
   - Resource group: the name of a new resource group **az3000700-LabRG**
   - Storage account: a name of a new storage account
   - File share: a name of a new file share

3. From the Cloud Shell pane, run the following to create a new Azure AD application that you will associate with the service principal you create in the subsequent steps of this task:

```
$password = 'Pa55w.rd1234'
```

```
$securePassword = ConvertTo-SecureString -Force -AsPlainText `
-String $password
```

```
$aadApp30007 = New-AzureRmADApplication -DisplayName 'aadApp30007' `
-HomePage 'http://aadApp30007' -IdentifierUris 'http://aadApp30007' `
-Password $securePassword
```

1. From the Cloud Shell pane, run the following to create a new Azure AD service principal associated with the application you created in the previous step:

   ```
   New-AzureRmADServicePrincipal `
   -ApplicationId $aadApp30007.ApplicationId.Guid
   ```

1. In the output of the **New-AzureRmADServicePrincipal** command, note the value of the **ApplicationId** property. You will need this in the next exercise of this lab.

2. From the Cloud Shell pane, run the following to identify the value of the **Id** property of the current Azure subscription and the value of the **TenantId** property of the Azure AD tenant associated with that subscription (you will also need them in the next exercise of this lab):

   ```
   Get-AzureRmSubscription
   ```

1. Close the Cloud Shell pane.

Task 4: Assign the Reader role to the Azure AD service principal

1. In the Azure portal, navigate to the blade displaying properties of your Azure subscription.

2. On the Azure subscription blade, click **Access control (IAM)**.

3. Assign the **Reader** role within the scope of the Azure subscription to the **aadApp30007** service principal.

Task 5: Register the Microsoft.EventGrid resource provider

1. In the Azure portal, in the Microsoft Edge window, reopen the **PowerShell** session within the **Cloud Shell**.

2. From the Cloud Shell pane, run the following to register the Microsoft.EventGrid resource provider:

   ```
   Register-AzureRmResourceProvider -ProviderNamespace Microsoft.EventGrid
   ```

1. Close the Cloud Shell pane.

Result: After you completed this exercise, you have created a storage account, a logic app that you will configure in the next exercise of this lab, and an Azure AD service principal that you will reference during that configuration.

Exercise 2: Configure Azure logic app to monitor changes to a resource group.

The main tasks for this exercise are as follows:

1. Add a trigger to the Azure logic app

2. Add an action to the Azure logic app

3. Identify the callback URL of the Azure logic app

4. Configure an event subscription

5. Test the logic app

Task 1: Add a trigger to the Azure logic app

1. In the the Azure portal, navigate to the **Logic App Designer** blade of the newly provisioned Azure logic app.

2. Click **Blank Logic App**. This will create a blank designer workspace and display a list of recent connectors and triggers to add to the workspace. It is likely to be empty.

3. Search for **Event Grid** triggers and, in the list of results, click the **When a resource event occurs (preview) Azure Event Grid** trigger to add it to the designer workspace.

4. In the **Azure Event Grid** tile, click the **Connect with Service Principal** link, specify the following values, and click **Create**:

   - Connection Name: **egc30007**
   - Client ID: the **ApplicationId** property you identified in the previous exercise
   - Client Secret: **Pa55w.rd1234**
   - Tenant: the **TenantId** property you identified in the previous exercise

5. In the **When a resource event occurs** tile, specify the following values:

   - Subscription: the subscription **Id** property you identified in the previous exercise
   - Resource Type: **Microsoft.Resources.resourceGroups**
   - Resource Name: **az3000701-LabRG**
   - Event Type Item - 1: **Microsoft.Resources.ResourceWriteSuccess**
   - Event Type Item - 2: **Microsoft.Resources.ResourceDeleteSuccess**

Task 2: Add an action to the Azure logic app

1. In the the Azure portal, on the Logic App Designer blade of the newly provisioned Azure logic app, click **+ New step**.

2. In the **Choose an action** pane, in the **Search connectors and actions** text box, type **Outlook**.

3. In the list of results, click **Outlook.com**.

4. In the list of actions for **Outlook.com**, click **Send an email**.

5. In the **Outlook.com** pane, click **Sign in**.

6. When prompted, authenticate by using the Microsoft Account you are using in this lab.

7. When prompted for the consent to grant Azure Logic App permissions to access Outlook resources, click **Yes** and when asked to confirm, click **Allow Access.**

8. In the **Send an email** pane, specify the following settings and click **Save**:

   – To: the name of your Microsoft Account

   – Subject: type **Resource updated:** and, in the **Dynamic Content** column to the right of the **Send an email** pane, click **Subject**.

   – Body: Type **Resource group:**, in the **Dynamic Content** column to the right of the **Send an email** pane, click **Topic**, type **Event type:**, in the **Dynamic Content** column to the right of the **Send an email** pane, click **Event Type**, type **Event ID:**, in the **Dynamic Content** column to the right of the **Send an email** pane, click **ID**, type **Event Time:**, and in the **Dynamic Content** column to the right of the **Send an email** pane, click **Event Time**.

Task 3: Identify the callback URL of the Azure logic app

1. In the Azure portal, navigate to the **logicapp3000701** Overview blade and, in the **Summary** section, click **See trigger history**.

2. On the **When_a_resource_event_occurs** blade, copy the value of the **Callback url [POST]** text box.

Task 4: Configure an event subscription

1. In the Azure portal, navigate to the **az3000701-LabRG** resource group and, in the vertical menu, click **Events.**

2. On the **az3000701-LabRG - Events** blade, click **Web Hook.**

3. On the **Create Event Subscription** blade, clear the **Subscribe to all event types** checkbox and, in the **Defined Event Types** drop down list, ensure that only the checkboxes next to the **Resource Write Success** and **Resource Delete Success** are selected.

4. In **Endpoint Details** select **Web Hook** in the **Endpoint Type** drop-down list, then click the **Select an endpoint** link.

5. On the **Select Web Hook** blade, in the **Subscriber Endpoint**, paste the value of the **Callback url [POST]** of the Azure logic app you copied in the previous task and click **Confirm Selection**.

6. In the **Name** text box within the **EVENT SUBSCRIPTION DETAILS** section, type **event-subscription-az3000701**.

7. Click **Create**.

Task 5: Test the logic app

1. In the Azure portal, navigate to the **az3000701-LabRG** resource group and, in the vertical menu, click **Overview.**

2. In the list of resources, click the Azure storage account you created in the first exercise.

3. On the storage account blade, in the vertical menu, click **Configuration**.

4. On the configuration blade, set the **Secure transfer required** setting to **Disabled**

5. Navigate to the **logicapp3000701** blade, click **Refresh**, and note that the **Runs history** includes the entry corresponding to configuration change of the Azure storage account.

6. Navigate to the inbox of the email account you configured in this exercise and verify that includes an email generated by the logic app.

Result: After you completed this exercise, you have configured an Azure logic app to monitor changes to a resource group.