

Online Lab - Integrating Azure Solution Components using Messaging Services

Topic: Deploying Messaging components to facilitate communication between Azure resources

Before we start

1. Ensure that you are logged in to your Windows 10 lab virtual machine using the following credentials:
 - Username: **Admin**
 - Password: **Pa55w.rd**
2. Review Taskbar located at the bottom of your Windows 10 desktop. The Taskbar contains the icons for the common applications you will use in the labs:
 - Microsoft Edge
 - File Explorer
 - [Visual Studio Code](#)
 - [Microsoft Azure Storage Explorer](#)
 - Bash on Ubuntu on Windows
 - Windows PowerShell

Note: You can also find shortcuts to these applications in the **Start Menu**.

Exercise 1: Deploy a Service Bus namespace

Task 1: Open the Azure portal

1. On the Taskbar, click the **Microsoft Edge** icon.
2. In the open browser window, navigate to the **Azure Portal** (<https://portal.azure.com>).
3. When prompted, authenticate with the user account account that has the owner role in the Azure subscription you will be using in this lab.

Task 2: Create a Service Bus namespace

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Service Bus** and press **Enter**.

3. On the **Everything** blade, in the search results, click **Service Bus**.
4. On the **Service Bus** blade, click the **Create** button.
5. On the **Create namespace** blade, perform the following tasks:
 - In the **Name** text box, enter a globally unique name.
 - In the **Pricing tier** drop-down list, select the **Basic** option.
 - Leave the **Subscription** drop-down list entry set to its default value.
 - In the **Resource group** section, ensure that the **Create new** option is selected and then, in the text box, type **AADesignLab1101-RG**.
 - In the **Location** drop-down list, select the Azure region to which you intend to deploy resources in this lab.
 - Click the **Create** button.
6. Wait for the provisioning to complete before you proceed to the next step.

Task 3: Create a Service Bus Queue

1. In the hub menu of the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab1101-RG**.
3. On the **AADesignLab1101-RG** blade, click the newly created Service Bus namespace.
4. On the Service Bus namespace blade, in the **ENTITIES** section, click **Queues**.
5. On the Service Bus namespace blade, click the **+ Queue** button.
6. In the **Create queue** pane, perform the following tasks:
 - In the **Name** text box, type **messages**.
 - Leave all remaining settings with their default values.
 - Click the **Create** button.

Task 4: Get Service Bus Connection String

1. Back on the Service Bus namespace blade, click **Shared access policies**.
2. On the Service Bus namespace blade, click the **RootManageSharedAccessKey** policy.
3. In the **SAS Policy: RootManageSharedAccessKey** pane, locate and record the value of the **Primary Connection String** field. You will use this value later in this lab.

Review: In this exercise, you created a new Service Bus namespace and recorded a connection string to access queues in the namespace.

Exercise 2: Create a logic app

Task 1: Create an Azure Storage account

1. In the upper left corner of the Azure portal, click **Create a resource**.
2. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Storage Account** and press **Enter**.
3. On the **Everything** blade, in the search results, click **Storage Account - blob, file, table, queue**.
4. On the **Storage Account - blob, file, table, queue** blade, click the **Create** button.
5. Create a storage account with the following settings:
 - Subscription: the subscription you are using in this lab.
 - Resource group: the resource group you created in the previous exercise.
 - Storage account name: any unique name consisting of a combination of between 3 and 24 characters and digits.
 - Location: the same Azure region you selected in the previous exercise.
 - Performance: **Standard**.
 - Account kind: **Storage (general purpose v1)**.
 - Replication: **Locally-redundant storage (LRS)**.
 - Secure transfer required: **Disabled**.
 - Allow access from: **All networks**.
 - Hierarchical namespace: **Disabled**.
6. Wait for the provisioning to complete before you proceed to the next step.
7. In the hub menu of the Azure portal, click **Resource groups**.
8. On the **Resource groups** blade, click **AADesignLab1101-RG**.
9. On the **AADesignLab1101-RG** blade, click the newly created Azure Storage account.
10. On the Storage account blade, click the **Blobs** tile.
11. On the Storage account blade, click the **+ Container** button.
12. In the **New container** pane, perform the following tasks:
 - In the **Name** text box, type **messageoutput**.
 - In the **Public access level** drop-down list, select the **Blob (anonymous read access for blobs only)** option.
 - Click the **OK** button.

Task 2: Create a logic app

1. At the top of the **New** blade, in the **Search the Marketplace** text box, type **Logic App** and press **Enter**.
2. On the **Everything** blade, in the search results, click **Logic App**.
3. On the **Logic App** blade, click the **Create** button.

4. On the **Create logic app** blade, perform the following tasks:
 - In the **Name** text box, type **ServiceBusWorkflow**.
 - Leave the **Subscription** drop-down list entry set to its default value.
 - In the **Resource group** section, select the **Use existing** option and then, in the drop-down list, select **AADesignLab1101-RG**.
 - In the **Location** drop-down list, select the same Azure region you chose in the previous task.
 - In the **Log Analytics** section, ensure that the **Off** button is selected.
 - Click the **Create** button.
5. Wait for the provisioning to complete before you proceed to the next task.

Task 3: Configure logic app steps.

1. In the hub menu in the Azure portal, click **Resource groups**.
2. On the **Resource groups** blade, click **AADesignLab1101-RG**.
3. On the **AADesignLab1101-RG** blade, click the entry representing the logic app you created in the previous task.
4. On the **Logic Apps Designer** blade, scroll down and click the **Blank Logic App** tile in the **Templates** section.
5. On the **Logic Apps Designer** blade, perform the following tasks:
 - In the **Search connectors and triggers** text box, type **Service Bus**.
 - In the search results, select the trigger named **When a message is received in a queue (auto-complete) - Service Bus**.
 - In the **Connection Name** text box, type **ServiceBusConnection**.
 - In the list of **Service Bus namespaces**, select the namespace you created earlier in this lab.
 - In the list of policies, select the **RootManageSharedAccessKey** policy.
 - Click the **Create** button.
6. In the **When a message is received in a queue (auto-complete)** step, perform the following tasks:
 - In the **Queue name** drop-down list, select the **messages** entry.
 - In the **Interval** text box, type **30**.
 - In the **Frequency** drop-down list, select the **Second** entry.
7. On the **Logic Apps Designer** blade, click the **+ New Step** button.
8. On the **Logic Apps Designer** blade, perform the following tasks:
 - In the **Search connectors and actions** text box, type **Storage blob**.
 - In the search results, select the action named **Create blob - Azure Blob Storage**.
 - In the **Connection Name** text box, type **StorageConnection**.

- In the list of *Storage accounts*, select the account you created earlier in this lab.
 - Click the **Create** button.
9. In the **Create Blob** step, perform the following tasks:
- In the **Folder path** text box, type **/messageoutput**.
 - In the **Blob name** text box, type **@concat(triggerBody()?['MessageId'], '.txt')**.
 - In the **Blob content** text box, type **@string(decodeBase64(triggerBody()?['ContentData']))**.
10. At the top of the **Logic Apps Designer** blade, click the **Save** button to persist your workflow.

Task 2: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open a new shell instance.

Note: The **Cloud Shell** icon is a symbol that is constructed of the combination of the *greater than* and *underscore* characters.

2. If this is your first time opening the **Cloud Shell** using your subscription, you will see a wizard to configure **Cloud Shell** for first-time usage. When prompted, in the **Welcome to Azure Cloud Shell** pane, click **Bash (Linux)**.

Note: If you do not see the configuration options for **Cloud Shell**, this is most likely because you are using an existing subscription with this course's labs. If so, proceed directly to the next task.

3. In the **You have no storage mounted** pane, click **Show advanced settings**, perform the following tasks:
- Leave the **Subscription** drop-down list entry set to its default value.
 - In the **Cloud Shell region** drop-down list, select the Azure region matching or near the location where you deployed resources in this lab.
 - In the **Resource group** section, select the **Use existing** option and then, in the drop-down list, select **AADesignLab1101-RG**.
 - In the **Storage account** section, ensure that the **Create new** option is selected and then, in the text box below, type a unique name consisting of a combination of between 3 and 24 characters and digits.
 - In the **File share** section, ensure that the **Create new** option is selected and then, in the text box below, type **cloudshell**.
 - Click the **Create storage** button.

4. Wait for the **Cloud Shell** to finish its first-time setup procedures before you proceed to the next task.

Task 4: Validate Logic App using Node.js

1. At the top of the portal, click the **Cloud Shell** icon to open a new shell instance.
2. At the **Cloud Shell** command prompt at the bottom of the portal, type in the following command and press **Enter** to install the **azure** package using NPM:

```
npm install azure
```

3. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to open the interactive node terminal:

```
node
```

4. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to import the **azure** module in Node:

```
var azure = require('azure');
```

5. At the **Cloud Shell** command prompt, type in the following command (replacing the placeholder `<Service Bus namespace connection string>` with the value of your url you recorded earlier in this lab) and press **Enter** to create a new variable for your Service Bus namespace connection string:

```
var connectionString = '<Service Bus namespace connection string>';
```

6. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to create a new client to connect to the Service Bus namespace:

```
var serviceBusService =  
azure.createServiceBusService(connectionString);
```

7. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to send a message to Service Bus namespace queue using the client.

```
serviceBusService.sendQueueMessage('messages', { body: 'Hello World' },  
function(error) { console.log(error) });
```

8. In the hub menu of the Azure portal, click **Resource groups**.
9. On the **Resource groups** blade, click **AADesignLab1101-RG**.
10. On the **AADesignLab1101-RG** blade, click the Azure Storage account you created earlier in this lab.

11. On the Storage account blade, click the **Blobs** tile.
12. On the Storage account container blade, click the **messageoutput** container.
13. Note the newly created blob in your container.

Review: In this exercise, you created a logic app that is triggered by messages from a queue in a Service Bus namespace.

Exercise 3: Remove lab resources

Task 1: Open Cloud Shell

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

```
az group list --query "[?starts_with(name, 'AADesignLab11')].name" --output tsv
```

3. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

Task 2: Delete resource groups

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name, 'AADesignLab11')].name" --output tsv | xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

Review: In this exercise, you removed the resources used in this lab.