CLOUD PLATFORMS AND ANALYTICS – WORKSHOP (WEEK 3)

Introduction to Azure Logic Apps

This workshop will give you a hands-on introduction to logic apps by walking you through a use case for creating an integration workflow with Azure Logic Apps using the Azure portal.

IMPORTANT:

- The services covered in this course are only a subset of a much larger family of Azure services. Similar outcomes can be achieved by leveraging other services and/or features not covered by this workshop. Specific business requirements may also ask for the use of different services or features not included in this workshop.
- Some concepts presented in this course can be quite complex and you may need to seek for more information from different sources (e.g. azure documentation) to compliment your understanding of the Azure services covered.

Document Structure

This document contains detailed step-by-step instructions on how to create and maintain a storage account using Azure portal and Python. It is recommended you carefully read the detailed description contained in this document for a successfully complete this workshop.

You will see the label **IMPORTANT** whenever a there is a critical step. Please pay close attention to the instructions given.

Table of Contents

Introduction

This workshop aims to provide a comprehensive understanding of creating and managing Azure Storage Accounts, specifically focusing on Blob Storage, and leveraging Azure Logic Apps to automate workflows.

The workshop will cover the essential steps to set up a new Azure Storage Account, emphasizing Blob Storage, which is Azure's object storage solution for the cloud. You will learn how to create, configure, and manage Blob Storage, understanding its significance in storing a vast array of unstructured data.

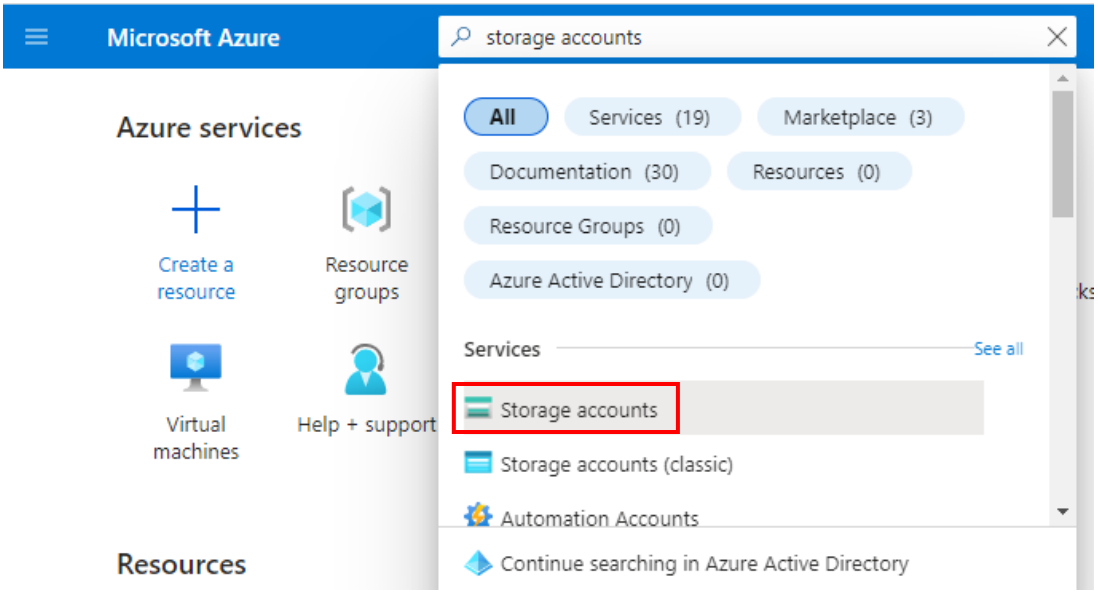
Furthermore, the workshop will delve into Azure Logic Apps, a powerful cloud service that allows you to automate and orchestrate tasks, workflows, and business processes. You will learn how to create a Logic App workflow designed to automate the process of saving email attachments directly to Blob Storage.

Prerequisites

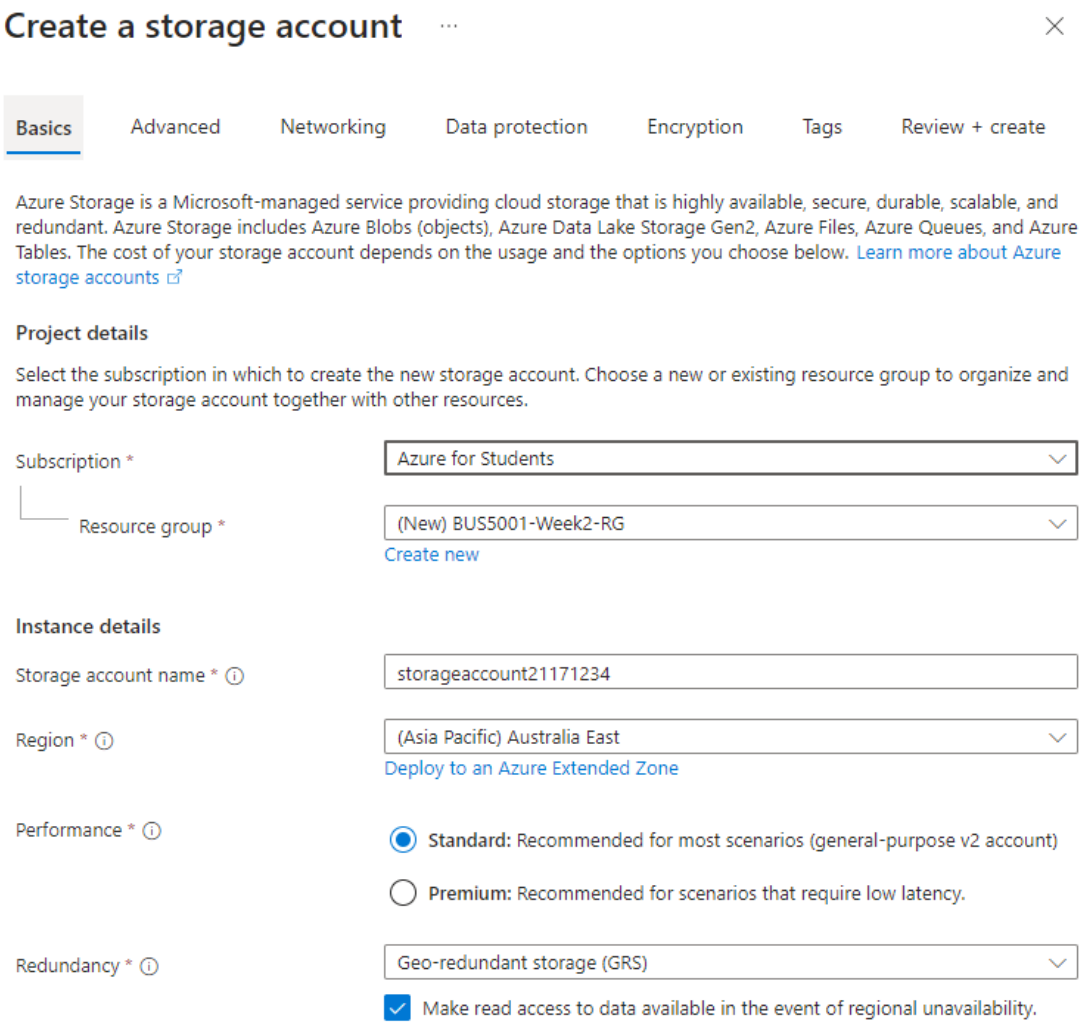
- An Azure account and subscription. If you don't have a subscription, sign up for a free Azure account using your La Trobe student account.

Create a storage account (Week 2 Recap)

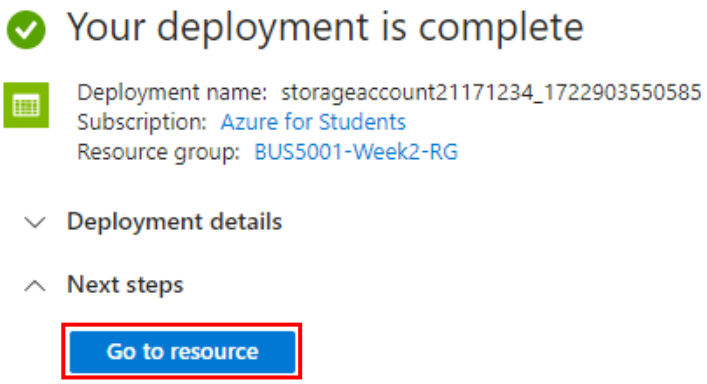
1. Sign in to the Azure portal with your Azure account.
2. In the Azure search box, enter 'storage account', and select Logic apps



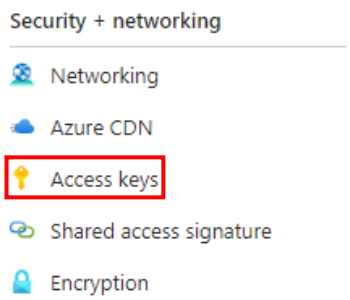
3. On the Storage accounts page, select **+ Create**.
4. On the **Create a storage account** pane, on the **Basics** tab, provide the following basic information about your logic app:
    - **Subscription**: Your Azure subscription name. *Ex: Azure for students*
    - **Resource Group**: Select **Create new** to create a new resource group: *Ex: BUS5001-Week2-RG.* (ensure the name is 1-90 characters long, contains only alphanumeric characters, periods, underscores, hyphens, and parentheses (no spaces), and is unique within the subscription)
    - **Storage account name:** *Ex: storageaccount<student_id>*. (ensure the name is 3-24 characters long, contains only lowercase letters and numbers, and is unique across all Azure)
5. Use default values for other settings. When you're done, your settings look similar to this version:



6. Select **Review**. On the validation page that appears, confirm all the information that you provided, and select **Create.**
7. After the deployment is completed. Select **Go to resource**.



8. On the left navigation panel, select **Access keys** under Security + networking.



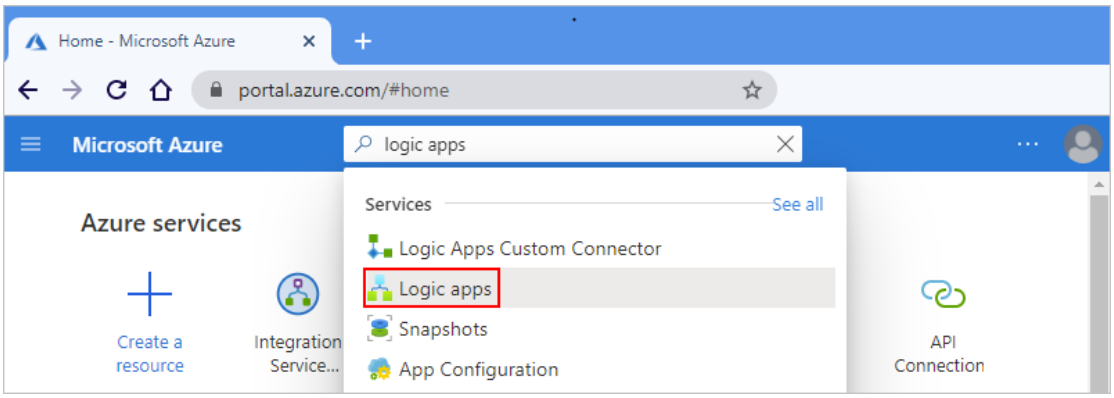
9. In a future step you will need the Storage account name and an access key. You will access them here.

Create a Container for email attachments

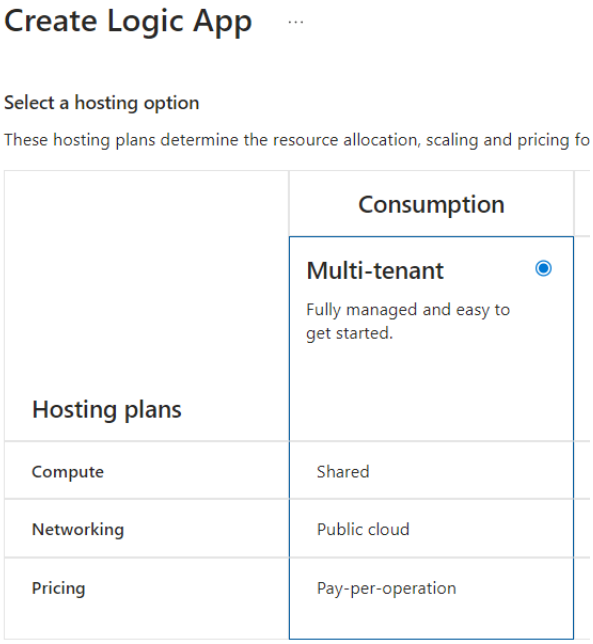
1. In the storage account, navigate to **Containers** under Data storage. Select **+ Container** to add a container named *emailattachments*. Access level can be default Private.

Create a Logic App resource

10. In the Azure search box, enter logic apps, and select Logic apps.

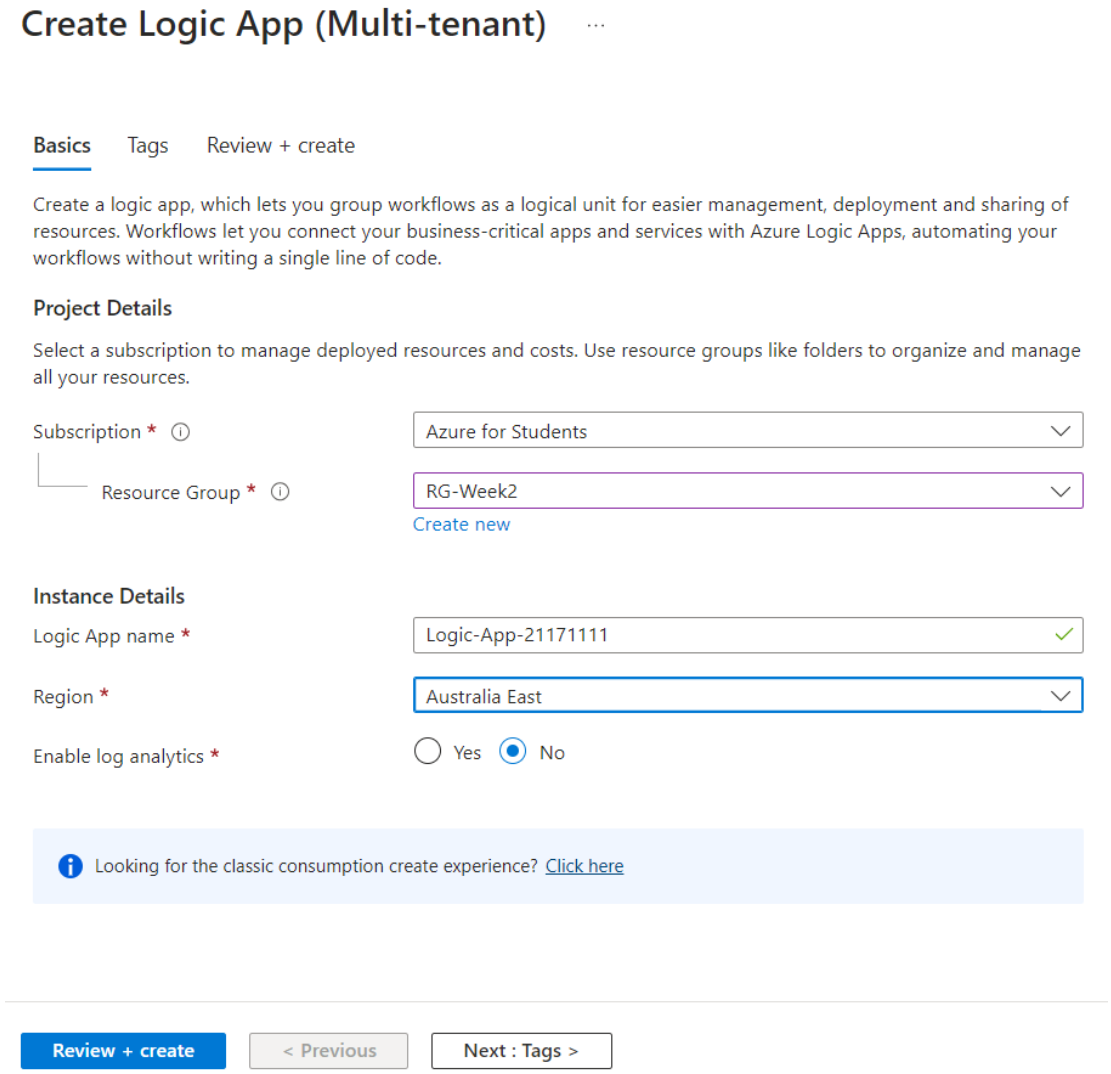


11. On the Logic apps page, select **+ Add**.
12. On the **Create Logic App** pane, select **Consumption, Multi-tenant** option and click **Select**.



13. On the **Create Logic App** pane, on the **Basics** tab, provide the following basic information about your logic app:
    a. **Subscription**: Your Azure subscription name. *Ex: Azure for students*
    b. **Resource Group**: Select the resource group that you used with the storage account.
    c. **Logic App name:** *Ex: storageaccount<student_id>*. (can contain only letters, numbers, hyphens (-), underscores (_), parentheses ((,)), and periods (.) and is unique across all regions). *Ex: Logic-App-<student_id>.*
    d. **Region:** Australia East
    e. **Enable log analytics:** No

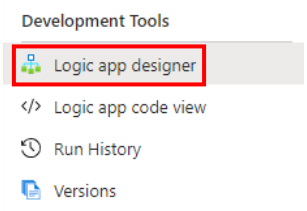
14. When you're done, your settings look similar to the following:



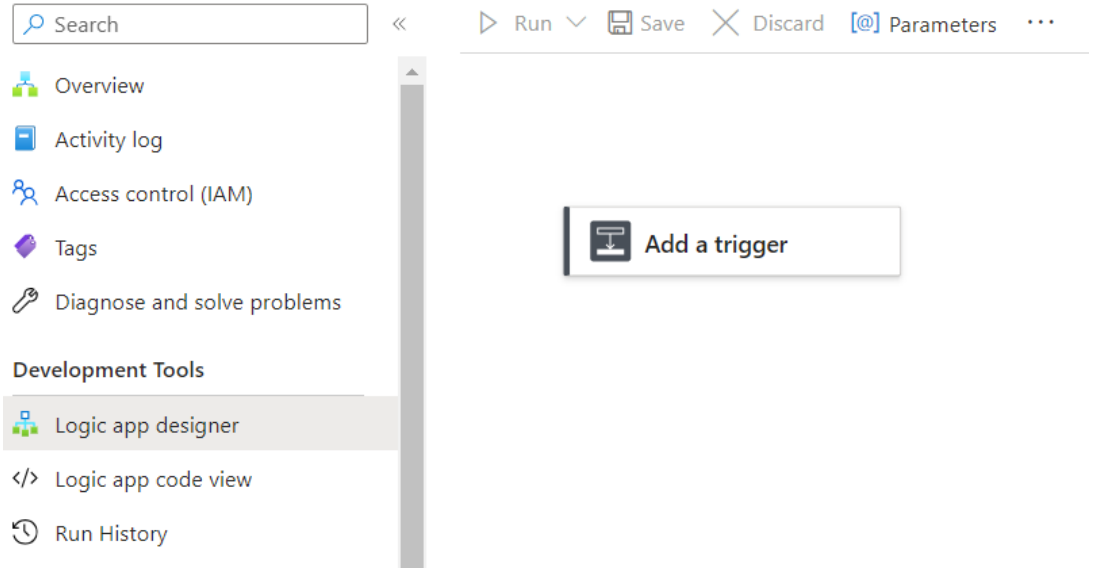
15. When you're ready, select **Review + Create**.
16. On the validation page that appears, confirm all the information that you provided, and select **Create**.

Select the blank template

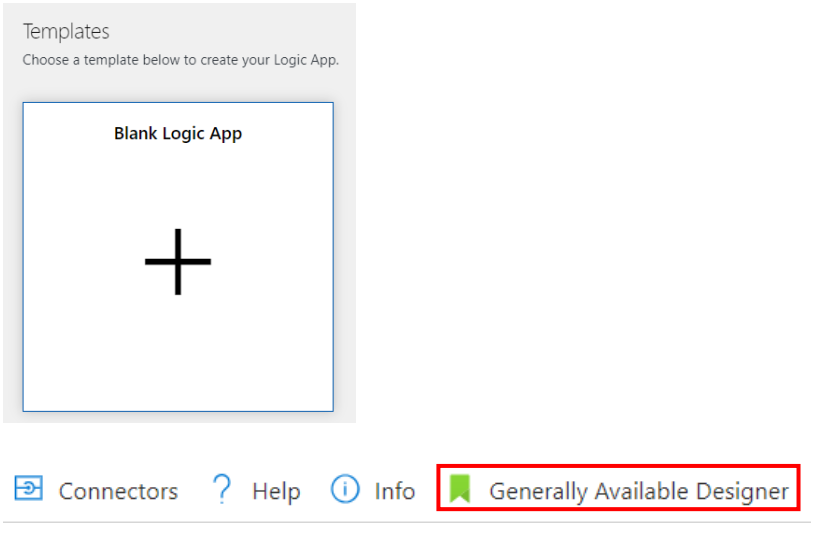
1. After Azure successfully deploys your app, select **Go to resource**. Or, find and select your logic app resource by typing the name in the Azure search box.
2. Select **Logic app designer** under **Development Tools** on the left navigation pane.



3. The designer should show an empty workflow where you can add the trigger.



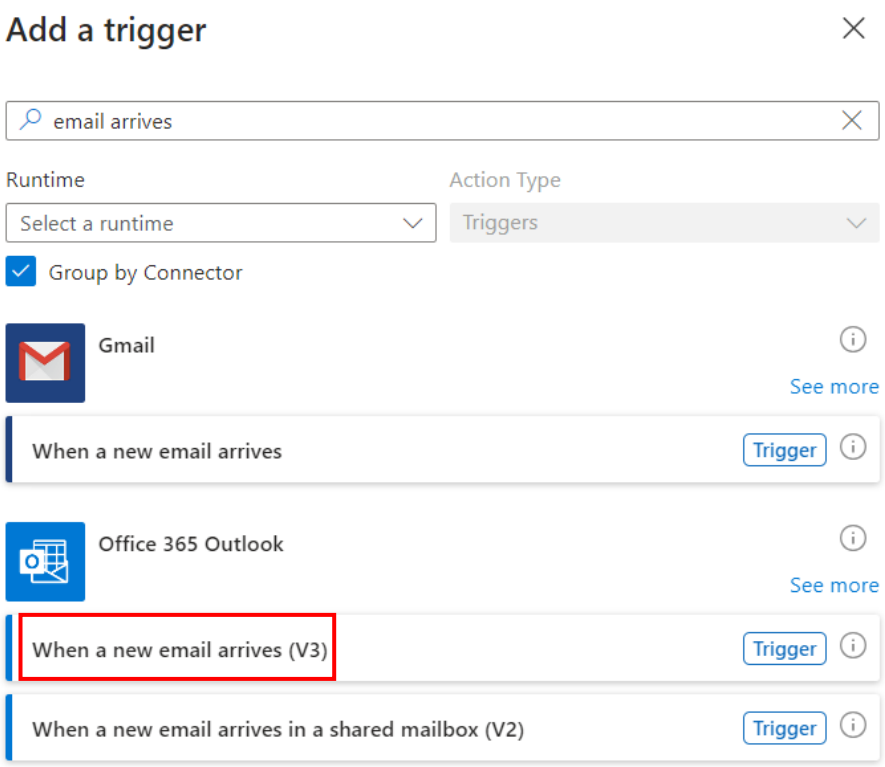
If not, scroll down and select **Blank Logic App** under Templates, Select **Generally Available Designer** from the upper pane.



Add the trigger

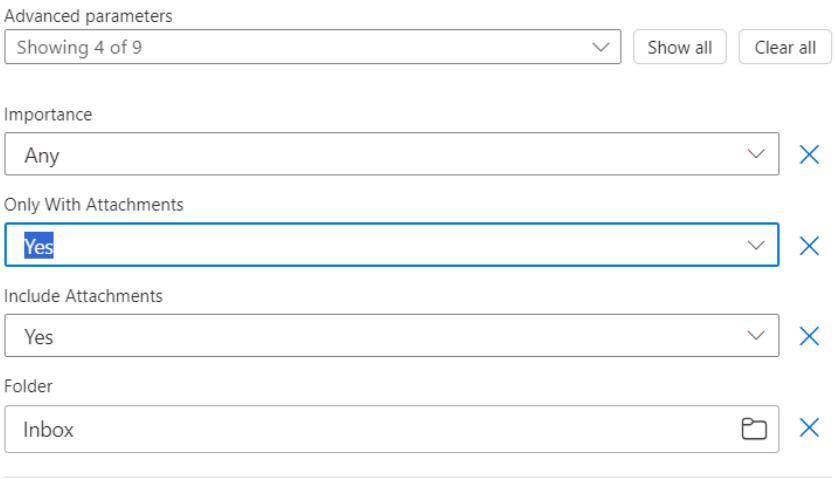
This workshop uses an Office 365 Outlook trigger that checks an email account. If a new email is received in the selected folder (ex: Inbox), the trigger fires, and a new workflow instance starts to run. If multiple new items exist between checks, the trigger fires for each item, and a separate new workflow instance runs for each item.

1. Select **Add a trigger** option.
2. In the search box, enter **'email arrives'**. Select the Office 365 Outlook trigger, **When a new email arrives (V3)**.



3. Select **Sign in** and sign in to your student email account.
4. In the **Parameters** pane, enter following details,

Property	**Required**	**Value**
Folder	Yes	Inbox
Importance	Yes	Any
Only with Attachments	Yes	Yes
Include Attachments	Yes	Yes



5. (Optional) In the Select **Advanced parameters** Dropdown and select **From**.

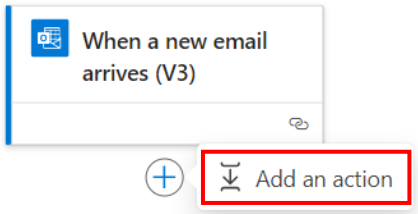
From	Yes	Email address that the email with the attachments is sent from. You can use your personal email address (Gmail) for this.

6. On the designer toolbar, select Save to save your logic app, which instantly goes live in the Azure portal.
   The trigger doesn't do anything other than check the inbox folder. So, you need to add an action that defines what happens when the trigger fires.

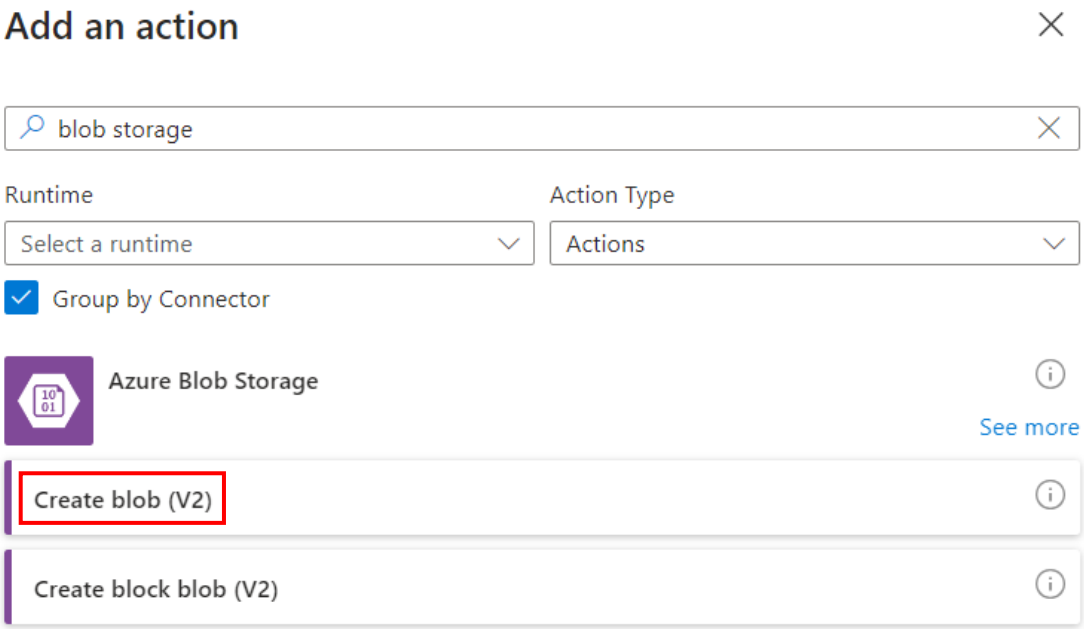
Add an Action

This workshop uses an Azure Blob storage action that create a blob for each attachment when the trigger fires for a new email.

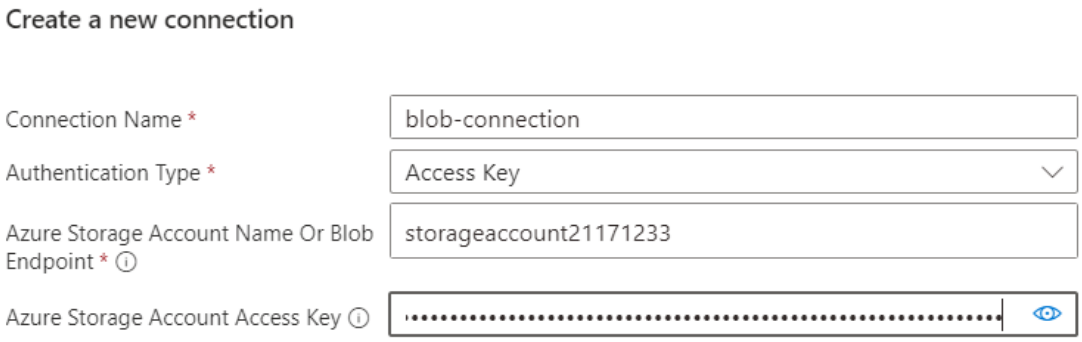
1. Under the **When a new email arrives (V3)** trigger, select **+**, then **Add Action.**



2. In the search box, enter **'blob storage'** and select **Create blob (V2)** under Azure Blob Storage.

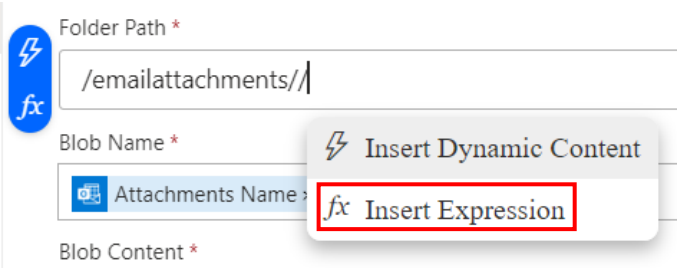


3. In the Azure Blob Storage action, change **Authentication type** to **Access Key**.
4. Enter following values to the fields,
    a. **Connection Name**: Name for the connection to the storage account. *Ex: blob-connection*.
    b. **Authentication Type**: **Access Key**
    c. **Azure Storage Account Name or Blob Endpoint**: Storage account name that you created. ('Create Storage Account' section step 9)
    d. **Azure Storage Account Access Key**: Storage Account Access Key. ('Create Storage Account' section step 9)

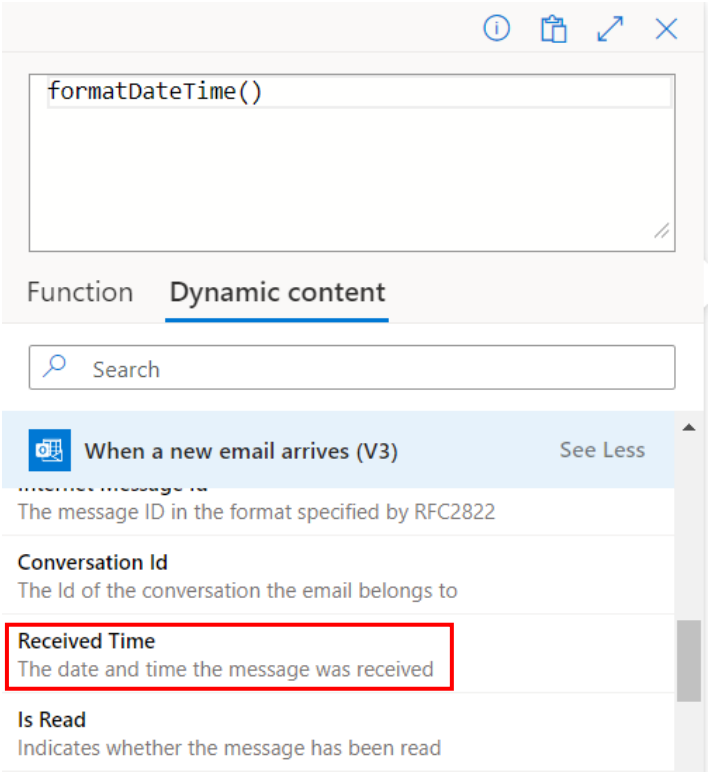


5. Select **Create**.
6. In the **Create blob (V2)** action,
    a. For **Storage account name or blob endpoint,** select the connection settings for the created storage account.
    b. For **Folder path**, enter */emailattachments*

After adding the created container name, **<u>add two '/' at the end</u>**.
Then select **Insert Expression**.



Enter 'formatDateTime' in the text box. Open paranthesis '(' and select **Dynamic content** tab. Under that, select **Received Time** (if you cannot see **Received Time**, then click **see more**).



Go to the end of the expression, inside the parenthesis, put a comma ',' and insert `'yyyy-MM-dd'`.
Final expression should look like this.

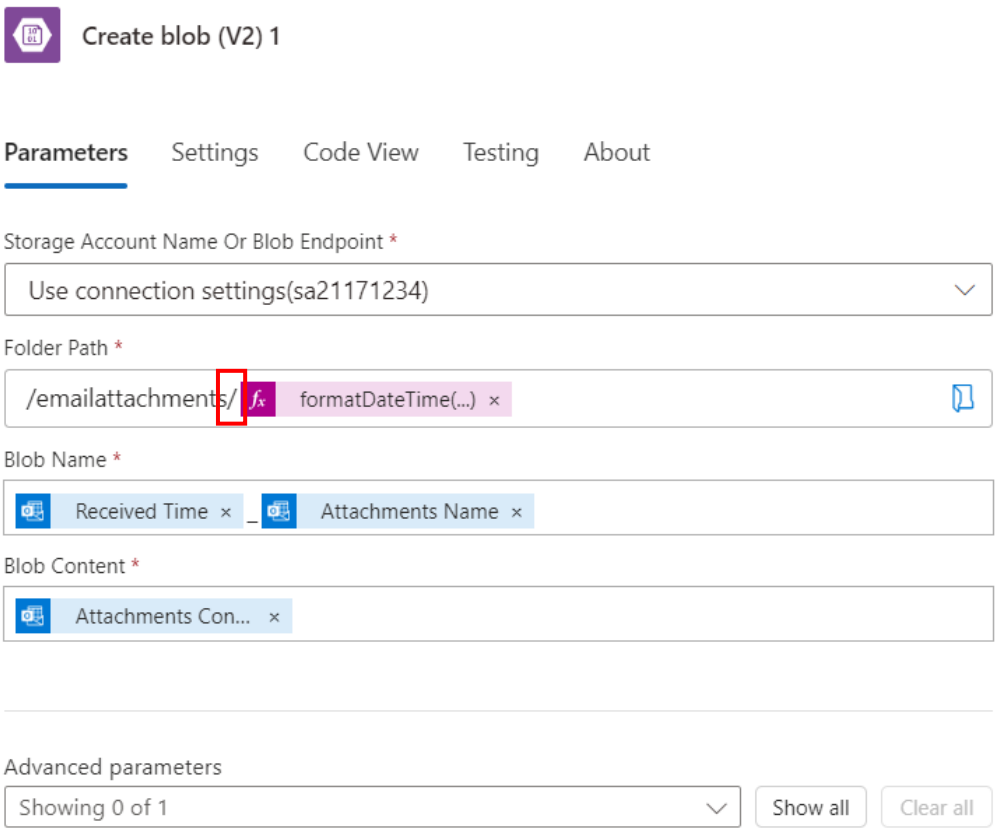
```
formatDateTime(triggerBody()?['receivedDateTime'], 'yyyy-MM-dd')
```

Click **ADD**.

c. For **Blob name**, from the dynamic content window select **Received Time**, add an underscore '_' and select **Attachments Name** from the dynamic content window.

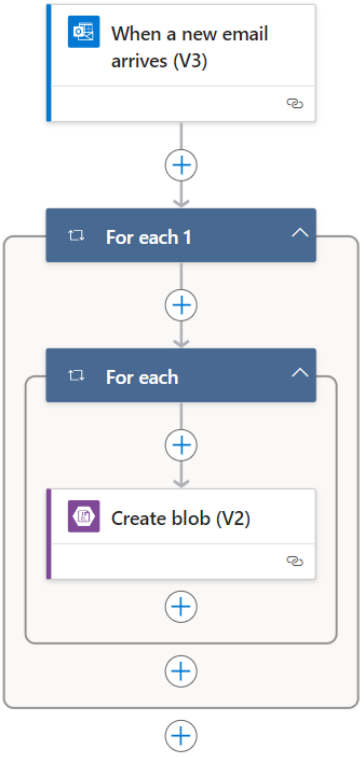
d. For **Blob content**, select **Attachments Content** from dynamic content window (Again, if you cannot see the property (Attachment Content) in the dynamic content window, click **see more**).

7. Final action window should look like the following.

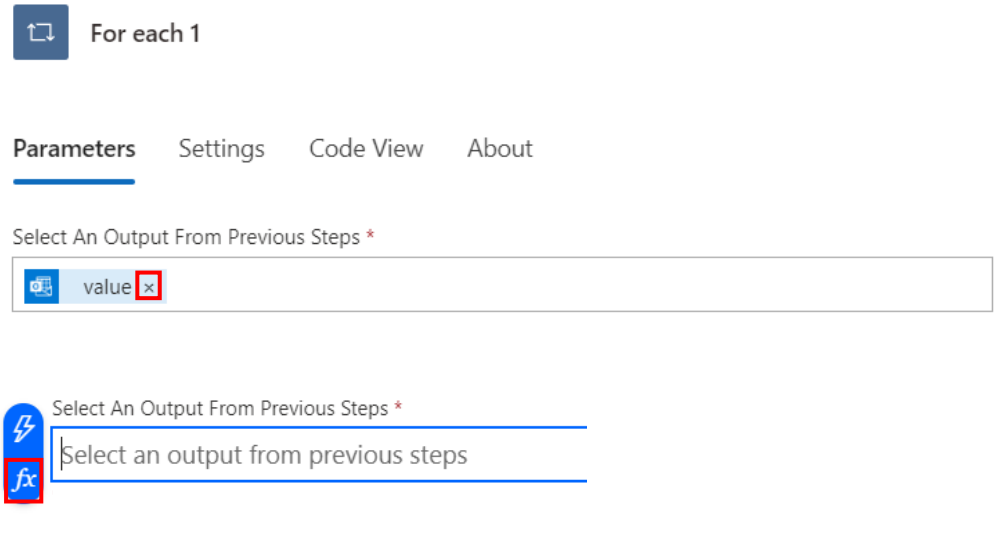


IMPORTANT: Make sure you have '/' after emailattachments, in Folder Path.

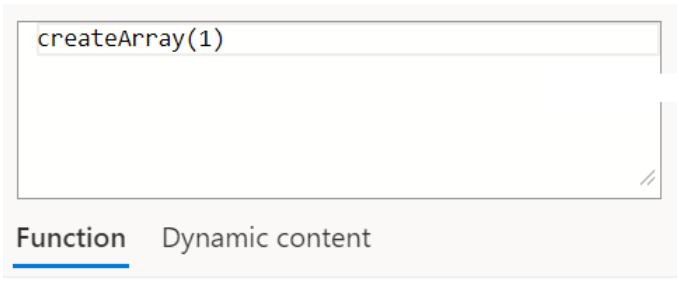
8. After that, you will see that there are two "For Each" controls have been added to the workflow.



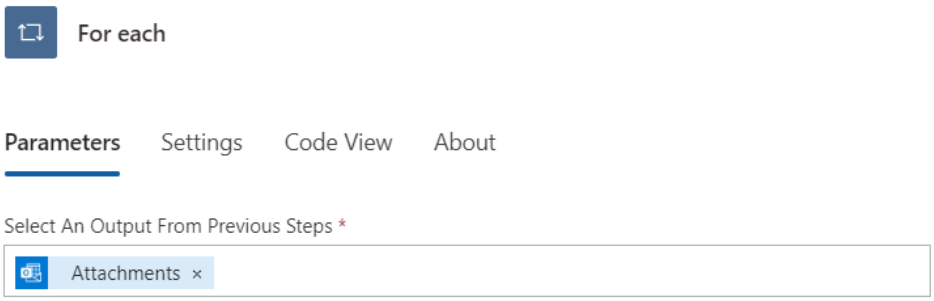
9. Select **For Each 1,** remove existing value, and change the **Select An Output From Previous Steps** to **Attachments**.



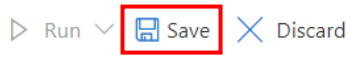
Add createArray(1) in the expression field. Click **Add**.



10. Select **For Each ,** <u>remove existing value</u>, and change the **Select An Output From Previous Steps** to **Attachments**.



11. Click **Save**.

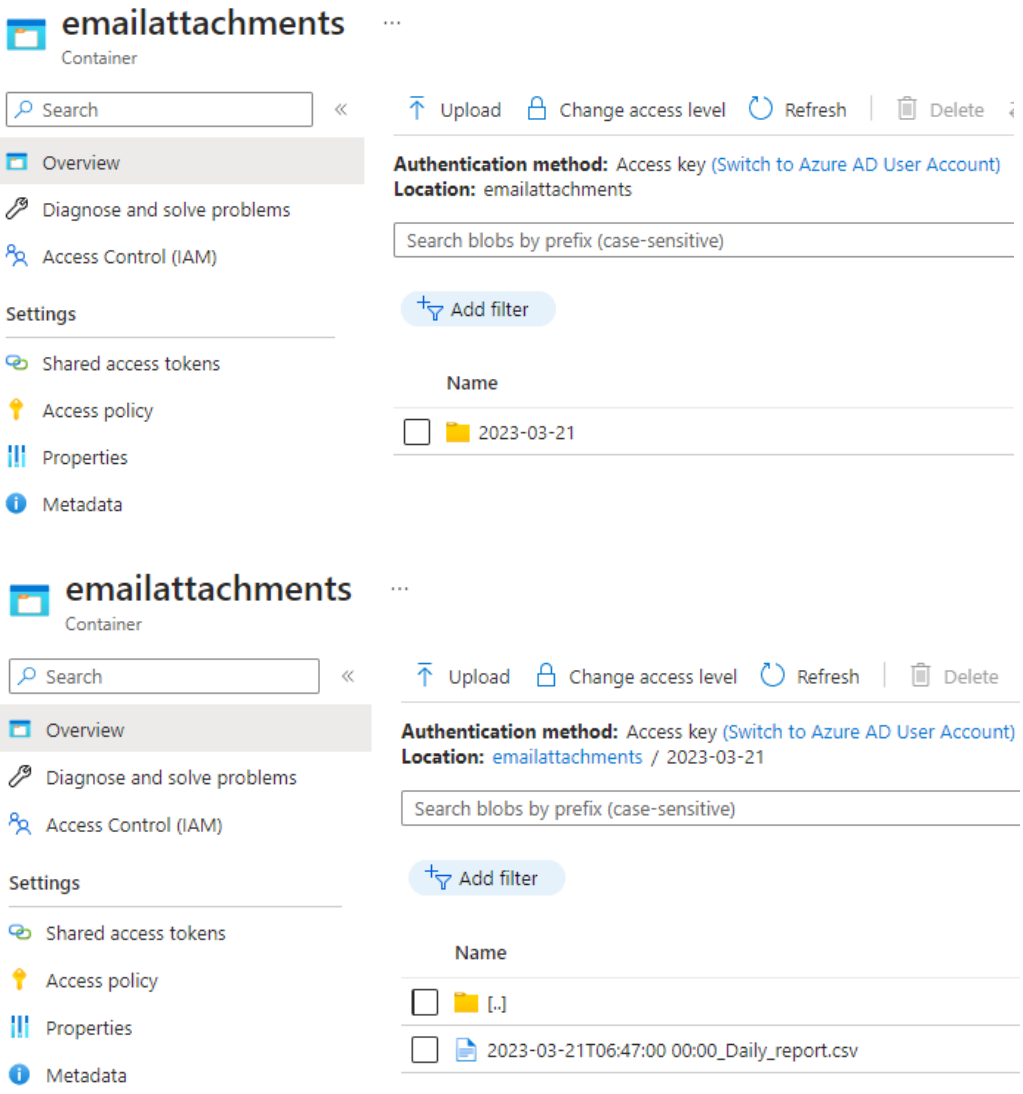


Run your workflow

To check that the workflow runs correctly, you can send an email with an attachment (ex: image, csv file) to your student email address (that you used to sign in to the trigger).

Note: **If you add 'From' field in the trigger, make sure you use the email address that you entered for the 'From' field in the trigger to send the email.**

If the logic app runs correctly, you should see a new folder and the attachment inside the container that you created within less than a minute after the email is received.



Clean up resources

IMPORTANT When you're done with this workshop, delete/disable the logic app resource and any related resources to prevent unnecessary cost.