LAB: Recognising people with Cognitive Services Face API

**Author:** Gavin Gregson | Mexia

# Objective

In this lab we are going to build a pair of Logic Apps, one to identify people from their photos and one to train the model. Images for both Logic Apps will be picked up from OneDrive, copied to intermediate Blob Storage, and then processed by Cognitive Services Face API.



# Prerequisites

* An Azure Subscription
* OneDrive
  + Two folders in OneDrive, an **IdentifyFaces** folder and a **TrainFaces** folder

# Recommended

* Azure Storage Explorer: <https://azure.microsoft.com/en-us/features/storage-explorer/>

Contents

[Log in to the Azure Portal 3](#_Toc509506460)

[Create a new dashboard 3](#_Toc509506461)

[Create a Face API resource and Resource Group 3](#_Toc509506462)

[Get your Face API key 4](#_Toc509506463)

[Create Storage Resources 5](#_Toc509506464)

[Create a Storage Account 5](#_Toc509506465)

[Create a Blob Container 6](#_Toc509506466)

[Create a Table 7](#_Toc509506467)

[Identifying a Face with Logic Apps 8](#_Toc509506468)

[What We’re Going to Build 8](#_Toc509506469)

[Create the Logic App 8](#_Toc509506470)

[Author the Logic App 9](#_Toc509506471)

[Add the OneDrive Trigger 9](#_Toc509506472)

[Copy the file to Blob storage 10](#_Toc509506473)

[Detect any Faces 11](#_Toc509506474)

[Identify any Faces 12](#_Toc509506475)

[Get the Identified Person’s Name 13](#_Toc509506476)

[Send an Email with the Persons Name and Picture 13](#_Toc509506477)

[Try it Out 15](#_Toc509506478)

[Training the Face API with Logic Apps 16](#_Toc509506479)

[What We’re Going to Build 16](#_Toc509506480)

[Create the Logic App 16](#_Toc509506481)

[Author the Logic App 17](#_Toc509506482)

[Add the OneDrive Trigger 17](#_Toc509506483)

[Copy the file to Blob storage 18](#_Toc509506484)

[Initialise a Variable to store the Person Group ID 19](#_Toc509506485)

[Initialise a Variable to store the Person's Name 19](#_Toc509506486)

[Initialise a Variable to store the Person's ID 19](#_Toc509506487)

[Get or Create a Person Group 20](#_Toc509506488)

[Get or Create a Person ID for a Person 22](#_Toc509506489)

[Add the Face Image to the Person 28](#_Toc509506490)

[Training the model 28](#_Toc509506491)

[Try it Out 29](#_Toc509506492)

# Log in to the Azure Portal

* Log in to <https://portal.azure.com>

# Create a new dashboard

* Create a new dashboard by clicking **+ New dashboard** and name it **GIBC**

# Create a Face API resource and Resource Group

* Click **+ Create a resource**
* Search for **Face API** and select the **Face API** resource from the search results  
  Everything 
  p Face 
  Results 
  NAME 
  Face API 
  VU Face Recogn 
  VU Face Recogn 
  - BYOL 
  PUBLISHER 
  Microsoft 
  VU LLC 
  VU LLC 
  CATEGORY 
  Al + Cognitive Services 
  Compute 
  Compute 
* Click the **Create** button at the bottom of the blade that appears  
  Face API 
  Add facial recognition capabilities to your app with the Face API. State-of-the-art algorithms detect 
  human faces in images, opening a range of facial capabilities to your apps. Face detection can also 
  identify attributes, including face landmarks (nose, eyes, etc.), gender, age, and other machine- 
  predicted facial features. 
  Once faces have been identified, the API can check to see if byo people in an image or images are 
  the same by using a confidence score, or compare it against a database to see a similar-looking or 
  identical face already exists. It can also organize similar faces into groups using shared visual traits. 
  Make facial recognition a key feature of your app with the Face API. 
  PUBLISHER 
  USEFUL LINKS 
  Create 
  Microsoft 
  More about Face API 
  Documentation 
  SDK 
  API reference 
  Pricing 
  Regional availability 
* Name the resource **gibc-face-api,** select a subscription and set the location to **Australia East**
* Set the pricing to the free **F0** tier
  + Note that you can only have 1 F0 tier Face API per subscription
* Under **Resource group** select **Create new** and name the resource group **gibc-face-lab**
* Make sure to tick **Pin to dashboard**
* Your Face API should look like this  
  Create 
  API 
  * Name 
  gibc-face api 
  * Subscription 
  Visual Studio Enterprise — 
  * Location 
  Australia East 
  MPN 
  * Pricing tier (View full pricing details) 
  FO (20 Calls per minute, 30K Calls per mon... v 
  * Resource group 
  C) use existing 
  Create new 
  gibc-face-lab 
  Pin to dashboard 
  Create 
  Automation options 
* Click **Create**, deployment should take around 30 seconds
* Click **Go to resource** from the alert that pops up

## Get your Face API key

* You'll be taken straight to the **Quick start** page. Click the **Keys** link under **1. Grab your key** and save it somewhere safe for use laterHome > gibc-face-api - Quick start 
  gibc-face-api - Quick start 
  Search [Ctrl +0 
  O Overview 
  Activty log 
  Access control (IA M) 
  Tags 
  Diagnose and solve problems 
  RESOURCE MANAGEMENT 
  Quick start 
  Pricing tier 
  @ Billing By Subscription 
  Properties 
  Congratulations! Your keys are ready. 
  Now explore the Quickstart guidance to get up and running with Face API. 
  Grab your keys 
  call to the Face API requires a subscription key. This key needs to be either passed through 
  a query string parameter or specified in the request header. You can find your keys in the API 
  resource 'Overview' or Keys' from the left menu. 
  Keys 
  Make an API call to endpoint 
  https://australiaeast.api.cognitive.microsoft.com/face/vl.c 
  Get in-depth information about each properties and methods of the API. Test your keys with the 
  built-in testing console without writing a single line of code. Once pu have the API running, you 
  can check your consumption and the API health on Azure portal in your API 'Overview'. 
  Face API reference 
  Realtime API usage 
  API metrics alert 
  Billing by subscription 
  Resource health status 

# Create Storage Resources

## Create a Storage Account

* Click **+ Create a resource** again
* Search for **Storage Account** and select the **Storage account - blob, file, table, queue** resource from the list  
  Home > New Marketplace > Everything 
  Marketplace 
  Everything 
  Compute 
  Networking 
  Storage 
  Web + Mobile 
  Databases 
  Data + Analytics 
  X 
  Everything 
  Y Filter 
  p Storage Account 
  Results 
  NAME 
  Storage account - blob, file, table, queue 
  Data Lake Store 
  ArcSight Logger 
  PUBLISHER 
  Microsoft 
  Microsoft 
  Hewlett Packard Enterprise 
  CATEGORY 
  S torage 
  S torage 
  Compute 
* Click the **Create** button at the bottom of the blade that appears
* Set a unique name for the storage account (3 to 24 lowercase alphanumeric characters)
* Since this is only for testing you can leave most settings at their defaults:
  + Deployment model: **Resource manager**
  + Account kind: **Storage (general purpose V1)**
  + Performance: **Standard**
  + Replication: **Read-access geo-redundant storage**
  + Secure transfer required: **Disabled**
  + Subscription: **<*your subscription>***
  + Virtual networks: **Disabled**
* For **Resource group** select **Use existing** and select the **gibc-face-lab** resource group that you created when you created the **Face API**
* Make sure to tick **Pin to dashboard**.
* Your Storage Account should look like this  
  Create storage account 
  The cost of your storage account depends on the 
  usage and the options you choose below. 
  Learn more 
  * Name O 
  gibcgpgOI 
  Deploymnent model O 
  Resource manager 
  Account kind O 
  .core.windows.net 
  Classic 
  Storage (general purpose VI) 
  Performance O 
  Standard 
  Premium 
  Replication O 
  Read-access geo-redundant storage (PA-. 
  * Secure transfer required O 
  Disabled 
  Enabled 
  * Subscription 
  Visual Studio Enterprise — MPN 
  * Resource group 
  o 
  Create new @ use existing 
  gibc-face-lab 
  * Location 
  Australia East 
  Virtual networks 
  Configure virtual neüorks O 
  Disabled 
  Enabled 
  Pin to dashboard 
  Create 
  Automation options 
* Click **Create**, deployment will take 30 - 60 seconds
* From the alert that pops up click **Go to resource**

## Create a Blob Container

* From the Storage Account **Overview** blade click **Blobs**gibcgpg01 
  StOØge 
  Search (Ctrl +0 
  Ove rview 
  Activty log 
  Access control (IA M) 
  Tags 
  Diagnose and solve problems 
  SETTINGS 
  Access keys 
  Configuration 
  Encryption 
  Shared access signature 
  Firewalls and virtual networks 
  Properties 
  Automation script 
  BLOB SERVICE 
  Containers 
  CORS 
  Custom domain 
  Azure CDN 
  Open in Explorer 
  Resource group (&Enge) 
  gibc-face-lab 
  Status 
  Delete storage account 
  Primary: Available, Secondary: Available 
  Location 
  Australia East, Australia Southeast 
  Subscription 
  Visual Studio Enterprise — MPN 
  Subscription ID 
  03587ca-ced3-4188-8c4e-6eb1774ad445 
  Services 
  Blobs 
  Object storage for understanding data 
  View metrics 
  Configure CORS rules 
  Setup custom domain 
  Tables 
  Tabular data storage 
  View metrics 
  Monitoring 
  Show data for last: 
  Total requests 
  hour 
  6 hours 
  12 hours 
  I day 
  No metric data could be loaded for the current timespan. 
  Performance 
  Standard 
  Replication 
  Read-access geo-redundant storage (RA-GRS) 
  Account kind 
  Storage (general purpose VI) 
  Files 
  File shares that use SM3 3.0 protocol 
  View metrics 
  Configure CORS rules 
  Scale apps depending on traffic 
  View metrics 
  7 days 
  Total egress 
  No metric data could be loaded for the current timespan. 
* Copy the **Primary blob service endpoint** (e.g. [https://*your-storage-account-name*.blob.core.windows.net](https://your-storage-account-name.blob.core.windows.net)) somewhere safe for later.
* Click **+ Container**
* Name the new container **faces** and set the **Public access level** to **Container**
  + We're going to be accessing blobs from the container by URL late, and for our lab purposes it's easiest to just make the container public. In a production scenario you would consider using something more secure, such as SAS tokens.
* Your container should look like this  
  Blob service 
  gibcgpg01 
  + Container Refresh 
  New container 
  Name 
  faces 
  Public access level O 
  Delete 
  Container (anonymous read access for containers and blobs) 
* Click **OK** and close the **Blob service** black using the **X** at the top-right

## Create a Table

* You should be back on the Storage Account's **Overview** blade
* Click **Tables**gibcgpg01 
  StOØge 
  Search (Ctrl +0 
  Ove rview 
  Activty log 
  Access control (IA M) 
  Tags 
  Diagnose and solve problems 
  SETTINGS 
  Access keys 
  Configuration 
  Encryption 
  Shared access signature 
  Firewalls and virtual networks 
  Properties 
  Automation script 
  BLOB SERVICE 
  Containers 
  CORS 
  Custom domain 
  Azure CDN 
  Open in Explorer 
  Resource group (&Enge) 
  gibc-face-lab 
  Status 
  Delete storage account 
  Primary: Available, Secondary: Available 
  Location 
  Australia East, Australia Southeast 
  Subscription 
  Visual Studio Enterprise — MPN 
  Subscription ID 
  03587ca-ced3-4188-8c4e-6eb1774ad445 
  Services 
  Blobs 
  Object storage for understanding data 
  View metrics 
  Configure CORS rules 
  Setup custom domain 
  Tables 
  Tabular data storage 
  View metrics 
  Monitoring 
  Show data for last: 
  Total requests 
  hour 
  6 hours 
  12 hours 
  I day 
  No metric data could be loaded for the current timespan. 
  Performance 
  Standard 
  Replication 
  Read-access geo-redundant storage (RA-GRS) 
  Account kind 
  Storage (general purpose VI) 
  Files 
  File shares that use SM3 3.0 protocol 
  View metrics 
  Configure CORS rules 
  Scale apps depending on traffic 
  View metrics 
  7 days 
  Total egress 
  No metric data could be loaded for the current timespan. 
* Ignore any prompts about Azure Cosmos DB, we're not going to use that for this lab
* Click **+ Table**
* Set the **Table name** to **PersonIds**
* Your table should look like this  
  Table service 
  gibcgpg01 
  Refresh Table 
  Add table 
  Table name 
  Personlds 
  Delete tables 
* Click **OK**

# Identifying a Face with Logic Apps

## What We’re Going to Build

* We’re going to build a Logic App which will look like this  
  When a file is created (Preview) 
  Create blob 
  Detect faces (Preview) 
  For each 
  Select an output from 
  previous steps 
  Body x 
  Get a person (Preview) 
  Send an email 
  Add an action 
  Add a condition 

## Create the Logic App

* In the Azure Portal click **+ Create a resource**
* Search for **Logic App** and select the **Logic App** resource from the list  
  Marketplace 
  Everything 
  Compute 
  Networking 
  Storage 
  Web + Mobile 
  Databases 
  Data + Analytics 
  Al + Cognitive Services 
  Internet of Things 
  X 
  Everything 
  Logic App 
  Results 
  NAME 
  Logic App 
  Logic Apps Management (Preview) 
  Logic Apps Custom Connector 
  Logic Apps 823 
  Sumo Logic for Azure Web Apps 
  PUBLISHER 
  Microsoft 
  Microsoft 
  Microsoft 
  Microsoft 
  Sumo Logic 
  CATEGORY 
  Web * Mobile 
  Monitoring * Manage... 
  Web * Mobile 
  Monitoring * Manage... 
  Compute 
* Click the **Create** button at the bottom of the blade that appears
* Set the **Name** of the logic app to **identify-faces**
* Set the **Subscription** to your subscription.
* Set **Resource group** to **Use existing** and select the **gibc-face-lab** resource group.
* Set **Location** to **Australia East**.
* Leave **Log Analytics** set to **Off**.
* Make sure you tick **Pin to dashboard**.
* Your Logic App should look like this  
  Name 
  id entify-faces 
  Subscription 
  Visual Studio Enterprise — MPN 
  Resource group O 
  C) Create new' use existing 
  gibc-föce•lab 
  Location 
  Australia East 
  Log Analytics 
  You can add triggers and actions 
  your Logic App after creation 
  Pin to dashboard 
  Create 
  Automation options 
* Click **Create**, deployment should take 30 - 60 seconds
* From the alert that pops up click **Go to resource**

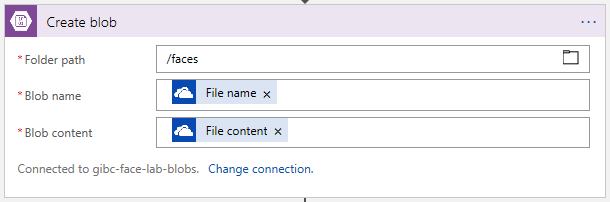
## Author the Logic App

* In the **Logic Apps Designer** blade that appears scroll down until you find the **Blank Logic App** template and click it

### Add the OneDrive Trigger

* The first step is to add a **Trigger**, so in the **Search all connectors and triggers** box search for **OneDrive** and select **OneDrive - When a file is created**  
  P OneDrive 
  Connectors 
  OneDrive 
  Excel Online 
  (Business) 
  Excel Online 
  (OneDrive) 
  OneDrive for 
  Business 
  OneNote 
  (Business) 
  See more 
  See more 
  Triggers (12) Actions (74) 
  OneDrive - When a file is created Preview 
  OneDrive - When a file is created (properties only) Preview 
  OneDrive - When a file is deleted (properties only) Preview 
  OneDrive - When a file is modified (properties only) Preview 
  OneDrive - When a file is modified 
  Preview 
  OneDrive for Business 
  OneDrive for Business 
  OneDrive far Business 
  When a file is created Preview 
  When a file is created (properties only) Preview 
  When a file is modified ro rties onl ) 
  Preview 
* Since you haven’t connected to OneDrive before you’ll need to sign in and create a connection, so click the **Sign In** button  
  OneDrive - When a file is created 
  Sign in to create a connection to OneDrive. 
  Sign in 
* Click **Yes** to the permissions request to complete the connection
* In the **Folder** field click the **folder icon** and select the folder in your OneDrive that you’ll put faces in for identification (**IdentifyFaces** if you created folders from the recommendations)
* In the action settings, set the **Interval** to **10 Seconds** for this lab so we don't have to wait too long for changes to be detected
* Your action should look like this  
  When a file is created (Preview) 
  Folder 
  /G18C/IdentifyFaces 
  Show advanced options v 
  How often do you want to check for items? 
  Interval 
  Connected to gpg@outlook.com. 
  F uen 
  Second 
  Change connection. 

### Copy the file to Blob storage

* Click **+ New step**, then **Add an action**
* Search for **Blob** and select the **Azure Blob Storage - Create blob** actionChoose an action 
  p Blob 
  Connectors 
  Azure 810b 
  Storage 
  Triggers (I) Actions (13) 
  Azure Blob Storage - Create blob 
  Azure Blob Storage - Create block blob Preview 
  Azure Blob Storage 
  Copy blob 
  See more 
  See more 
* First, we'll have to create a connection to the Storage Account we created earlier
  + Name the connection **gibc-face-lab-blobs**
  + Select the storage account you created earlier
* The connection should look like this  
  Create blab 
  *Connection Name 
  gibc-face lab-blobs 
  * Storage Account 
  gibcgpgOI 
  Resource Group 
  gibc-face-lab 
  Create 
  australiaeast 
  Manually enter connection information 
* Click **Create**
* In the action settings click the **folder icon** in the **Folder path** field and select the **faces** container you created earlier
* Click in the **Blob name** field and from the popup select **File name**
* Click in the **Blob content** field and from the popup select **File content**
* Your action should look like this  
  

### Detect any Faces

* Before we can identify a face we first need to detect any faces in the input image
* Click **+ New step**, then **Add an action**
* Search for **Detect Faces** and select the **Face API - Detect faces** action  
  Choose an action 
  p Detect Faces 
  Connectors 
  Face API 
  Triggers (O) Actions (I I) 
  See more 
  See more 
  Face API 
  Face API 
  Face API 
  - Detect faces Preview 
  - Add a person face Preview 
  - Add face to face list Preview 
* We’ll need to create a connection to the Face API
* Name the connection **face-api**
* For this GIBC 2018 lab we're initially going to use a pre-trained Face API instance that has already been created, then later you'll create and train your own instance, so for now set the API Key to  
  333fdfc41fcd4e8095b74a644aab72a5
* Set the **Site URL** to <https://australiaeast.api.cognitive.microsoft.com>
* Your action should look like this  
  Face API - Detect faces 
  *Connection Name 
  face-epi 
  * API Key 
  Site URL 
  https.//australiaeastapi.cognitive.microsoftcom 
  Create 
* Click **Create**
* In the action settings click in the **Image Url** field and enter the Blob endpoint you saved earlier without a trailing slash, e.g. <https://gibcgpg01.blob.core.windows.net>, then with the cursor at the end of the URL click **Path** from the popup
* Your action should look like this  
  Connected to gibc-face-lab-blobs. Change connection. 
  Detect faces 
  (P review) 
  * Image Orl 
  Connected to face-api. 
  P ath 
  https://gibcgpgOI.blob.core windows.net 
  Change connection. 
  + New step 
  Add dynamic content 
  Dynamic content Expression 
  Search dynamic content 
  A boolean value (true, false) to indicate whether or not th... 
  LastModified 
  The date and time the file or folder was last modified. 
  MediaType 
  The media type of the file or folder. 
  The name of the file or folder. 
  The path of the file or folder. 
  The size of the file or folder. 
  When a file is created 
  File content 
  The content of the file. 
  File content type 
  The content type of the file or folder. 
* Note that Detect Faces will return more than one Face ID if the image you provide has multiple faces in it

### Identify any Faces

* Now that we have detected a face in the image we need to identify it. Unfortunately, there isn't an action for this yet in Logic Apps, but it's very easy for us to make the REST API call ourselves.
* Because we might have more than one face detected we'll add a **For Each** loop to iterate over each Face ID.
* Click **+ New step,** but this time click **... More** and select **Add a for each**Add an action 
  + New step 
  Add a condition 
  More 
  Add a switch case 
  Add a for each 
  Add a do until 
  Add a scope 
* Click in the **Select an output from previous steps** field and select **Body** from the popup  
  Detect faces (Preview) 
  For each 
  * Select an output from 
  previous steps 
  Add an action 
  Add dynamic content 
  Dynamic content Expression 
  p Search dynamic content 
  Detect faces 
  Body 
  Add a condition 
  + New step 
* Within the **For each** action click **Add an action**, search for **HTTP**, and select **HTTP – HTTP**Choose an action 
  HTTP 
  Connectors 
  HTT P 
  Azure Kusto 
  Content 
  Moderator 
  HTTP with 
  Azure AD 
  Instagram 
  Nexmo 
  Triggers (O) Actions (44) 
  See more 
  Pitney 
  Bowes Data 
  See more 
  Azure Kusto 
  Azure Kusto 
  Run control command and visualize results Preview 
  Azure Kusto - Run query and list results Preview 
  Run query and visualize results Preview 
  Content Moderator 
  Preview 
  Content Moderator 
  content Preview 
  Create Reviews for Reviewers in your moderation 
  Detect Language of a given text input 
  H TTP 
  H TTP 
  HTTP 
  HTTp 
  HI-rp + Swagger 
  HTTP Webhook 
  Cancel 
* Set **Method** to **POST**
* Set the **Uri** to <https://australiaeast.api.cognitive.microsoft.com/face/v1.0/identify>
* Set the first header **key** to **Ocp-Apim-Subscription-Key** and the **value** to 333fdfc41fcd4e8095b74a644aab72a5
* Set the **Body** to the following JSON content:  
  **{  
   "faceIds": [""],  
   "personGroupId": "gotg"  
  }**
* Put the cursor in between the empty quotes **""** after **"faceIds"** and select **Face Id** from the popup, the **Body** field should look like thisfaceld 
  'facelds": 
  Body 
  personGroupId": "gotg" 
  Show advanced options v 
  Add dynamic content 
  p Search dynamic content 
  Body 
  Face Id 
  Unique faceld of the detected face. Will expire 24 hours a... 
  Face Rectangle 
* Your action should look like this  
  HTTp 
  Method 
  Headers 
  Body 
  Show advanced options v 
  POST 
  https://australiaeast.api.cognitive.microsoft.com/face/vI W identify 
  Ocp-Apim-Su bscription-Key 
  Enter key 
  faceld X 
  'facelds": 
  personGroupId": "gotg" 
  333fdfc41fcd4e8095b74a644aa 
  b72a5 
  Enter value 
  Add dynamic content 
* The response we receive will contain a **personId** for the person that best matches the face provided if the match exceeds a certain threshold (0.5 by default), we will use this to look up the person's details

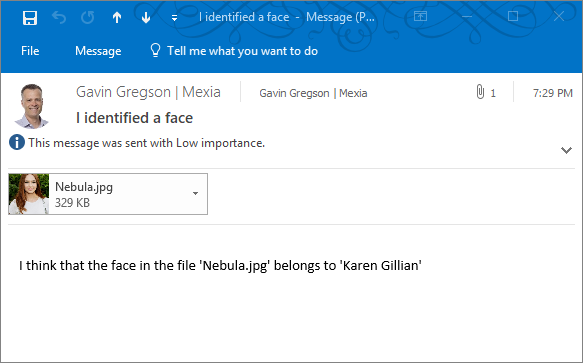
### Get the Identified Person’s Name

* In the **For each** loop click **Add an action**, search for **Get Person**, then select the **Face API - Get a person** action  
  Choose an action 
  Get Person 
  Connectors 
  Excel Online 
  (OneDrive) 
  Triggers (O) 
  Face API 
  Actions (10) 
  Mandrill 
  Workday 
  Excel Online (OneDrive) - Get a row Preview 
  Excel Online (OneDrive) - Get worksheets Preview 
  Excel Online (OneDrive) - Get tables Preview 
  Face API 
  - Get a person 
  Preview 
  Face API 
  - Get a person face Preview 
  Face API 
  - Get a person group Preview 
  Mandrill - Current user info 
  Mandrill - List scheduled messa e 
  See more 
  See more 
  Cancel 
* In the action settings set the **Person Group Id** to **gotg**
* Click in the **Person Id** field, click the **Expression** tab in the popup and set the expression to:  
  body('HTTP')[0]['candidates'][0]['personId']
* Your action should look like this  
  Get a person 
  (Preview) 
  Dynamic content Expression 
  body( 'HTTP' ' candidates persor• 
  Person Group Id 
  gotg 
  body(...) x 
  O 
  Add dynamic content 
  Update 
  String functions 
  text_2?, 
  Combines any number of strings together 
  See mare 
  Connected to face-api. Change connection. 

### Send an Email with the Persons Name and Picture

* Finally, we're going to email ourselves when we detect a face successfully. NOTE: This assumes that you have an **Office 365 Outlook** account to use. If you want to use Outlook.com, Gmail, or another service then you can search for the appropriate service and use that connector instead.
* In the **For each** action click **Add an action**, search for **Outlook** and select the **Office 365 Outlook - Send an email** actionChoose an action 
  Outlook 
  Connectors 
  Office 365 
  Outlook 
  Outlook 
  Customer 
  Outlook 
  Tasks 
  Outlook.co... 
  Triggers (18) Actions (72) 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  Office 365 Outlook 
  - Create contact 
  - Create event (V2) 
  - Send an email 
  - Send approval email 
  - Delete contact 
  - Delete email 
  - Delete event 
  ort email 
  Preview 
  See more 
  See more 
  Cancel 
* Sign in to your account to create a connection  
  Office 365 Outlook - Send an email 
  Sign in to create a connection to Office 365 Outlook. 
  Sign in 
* Set **To** to *your email address*
* Set **Subject** to **I identified a face**
* Set **Body** to **I think that the face in the file '' belongs to ''**
* Place the cursor between the empty quotes **''** after the word **file** and select **File name** from the popup under the*When a file is created* category (you'll probably need to scroll to the bottom to find it)
* Place the cursor between the empty quotes **''** after the word **to** and select **Name** from the popup under the *Get a person* category
* Click **Show advanced options**
* Click in the **Attachments Name - 1** field and select **File name** from the popup under the **When a file is created** category.
* Click in the **Attachments Content - 1** field and select **File contents** from the popup, also under the **When a file is created** category.
* Your action should look like this  
  Send an email 
  Subject 
  Body 
  From (Send as) 
  Attachments Name 
  Attachments Content 
  Importance 
  Is HTML 
  gaving@mexia.com.au 
  I identified a face 
  I think that the face in the file 
  File namex 
  belongs to 
  Name X 
  Add dynamic content 
  Email address to send mail from (requires •Send as" or •Send on behalf off pe... 
  Speci%' email addresses separated by semicolons like someone@contoso.com 
  Speci%' email addresses separated by semicolons like someone@contoso.com 
  File contentx 
  Importance 
  Is Html? 
  Hide advanced options 
  Connected to gaving@mexia.com.au.Change connection. 

### Try it Out

* Drop a photo that the Face API has been trained on into the folder the OneDrive trigger is connected to and if all goes well you should get an email in under 30 seconds with the recognised name and the triggering photo attached, e.g.  
  

# Training the Face API with Logic Apps

## What We’re Going to Build

* We’re going to build a Logic App which will look like this  
  When a file is created (Preview) 
  Create blob 
  Initialize person-group-id variable 
  Initialize person-name variable 
  Initialize person-id variable 
  Get person group scope 
  Get a person group (Preview) 
  O 
  Create a person group (Preview) 
  Get person ID scope 
  O 
  {x} 
  Create a person (Preview) 
  Set person-id variable from create 
  Insert Entity (Preview) 
  Add an action 
  Get entity (Preview) 
  Compose 
  Add an action 
  Add a person face (Preview) 
  HTTp 
  Add a condition 
  Set person-id variable from entity 
  O 
  Add a condition 
  + New step 

## Create the Logic App

* In the Azure Portal click **+ Create a resource**
* Search for **Logic App** and select the **Logic App** resource from the list
* Click the **Create** button at the bottom of the blade that appears
* Set the **Name** of the logic app to **train-faces**
* Set the **Subscription** to your subscription.
* Set **Resource group** to **Use existing** and select the **gibc-face-lab** resource group.
* Set **Location** to **Australia East**.
* Leave **Log Analytics** set to **Off**.
* Make sure you tick **Pin to dashboard**.
* Your Logic App should look like this  
  Create logic app 
  Logic App 
  * Name 
  train-faces 
  * Subscription 
  Visual Studio Enterprise — MPN 
  * Resource group O 
  o 
  Create new @ use existing 
  gibc-face-lab 
  Location 
  Australia East 
  Log Analytics O 
  You can add triggers and actions 
  your Logic App after creation. 
  Pin to dashboard 
  Create 
  Automation options 
* Click **Create**, deployment should take 30 - 60 seconds
* From the alert that pops up click **Go to resource**

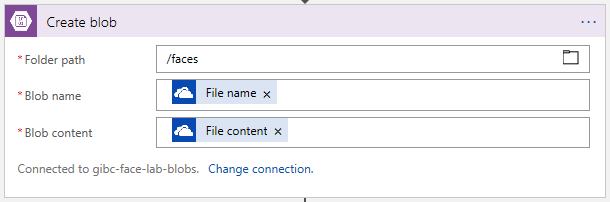
## Author the Logic App

* In the **Logic Apps Designer** blade that appears scroll down until you find the **Blank Logic App** template and click it

### Add the OneDrive Trigger

* The first step is to add a **Trigger**, so in the **Search all connectors and triggers** box search for **OneDrive** and select **OneDrive - When a file is created**
* You already have a connection to OneDrive so the action should use that automatically, but if you are presented with a Sign In option then just sign in and create a new connection
* In the **Folder** field click the **folder icon** and select the folder in your OneDrive that you’ll put faces in for identification (**TrainFaces** if you created folders from the recommendations)
* In the action settings, set the **Interval** to **10 Seconds** for this lab so we don't have to wait too long for changes to be detected
* Your action should look like this  
  When a file is created (Preview) 
  Folder 
  /G18C,TrainFaces 
  Show advanced options v 
  How often do you want to check for items? 
  Interval 
  Connected to gpg@outlook.com. 
  F uen 
  Second 
  Change connection. 

### Copy the file to Blob storage

* Click **+ New step**, then **Add an action**
* Search for **Blob** and select the **Azure Blob Storage - Create blob** actionChoose an action 
  p Blob 
  Connectors 
  Azure 810b 
  Storage 
  Triggers (I) Actions (13) 
  Azure Blob Storage - Create blob 
  Azure Blob Storage - Create block blob Preview 
  Azure Blob Storage 
  Copy blob 
  See more 
  See more 
* First, we'll have to create a connection to the Storage Account we created earlier
  + Name the connection **gibc-face-lab-blobs**
  + Select the storage account you created earlier
* The connection should look like this  
  Create blab 
  *Connection Name 
  gibc-face lab-blobs 
  * Storage Account 
  gibcgpgOI 
  Resource Group 
  gibc-face-lab 
  Create 
  australiaeast 
  Manually enter connection information 
* Click **Create**
* In the action settings click the **folder icon** in the **Folder path** field and select the **faces** container you created earlier
* Click in the **Blob name** field and from the popup select **File name**
* Click in the **Blob content** field and from the popup select **File content**
* Your action should look like this  
  

### Initialise a Variable to store the Person Group ID

* Click **+ New step**, then **Add an action**
* Search for **Variables** and select the **Variables - Initialize variable** action  
  Choose an action 
  p Variables 
  Connectors 
  {x} 
  Variables 
  Triggers (O) Actions (6) 
  Variables - Append to array variable 
  Variables - Append to string variable 
  Variables - Decrement variable 
  Variables - Increment variable 
  Variables - Initialize variable 
  Variables - Set variable 
  TELL US WHAT YOU NEED 
  See more 
  See more 
  O 
  Help us decide which connectors and triggers to add next with UserVoice 
  Cancel 
* Set the **Name** to **person-group-id**
* Set the **Type** to **String**
* Set the **Value** to **gotg**
* It's good practice to name actions meaningfully, particularly if there are going to multiple of the same type, so click on the **Ellipsis (...) menu**, select **Rename** and name this Action **Initialize person-group-id variable**
* Your action should look like this  
  {x} 
  Initialize person-group-id variable 
  * Name 
  * Type 
  Value 
  person-group-id 
  String 
  got g 

### Initialise a Variable to store the Person's Name

* As before, Click **+ New step**, then **Add an action**, search for **Variables** and select the **Variables - Initialize variable** action
* Set the **Name** to **person-name**
* Set the **Type** to **String**
* Click in the **Value** field and in the popup click the **Expression** tab
* Set the Expression to  
  split(triggerOutputs()['headers']['x-ms-file-name'], '-')[0]
* Click on the **Ellipsis (...) menu**, select **Rename** and name this Action **Initialize person-name variable**
* Your action should look like this  
  Initialize person-name variable 
  Dynamic content Expression 
  headers ' ] [ ' x-ms-fil'' 
  * Name 
  * Type 
  Value 
  person-name 
  String 
  Add dynamic content 
  Update 
  String functions 
  See more 

### Initialise a Variable to store the Person's ID

* Once more click **+ New step**, then **Add an action**, search for **Variables** and select the **Variables - Initialize variable** action
* Set the **Name** to **person-id**
* Set the **Type** to **String**
* Leave the **Value** field blank
* Click on the **Ellipsis (...) menu**, select **Rename** and name this Action **Initialize person-id variable**
* Your action should look like this  
  Initialize person-id variable 
  * Name 
  * Type 
  Value 
  pe rson-id 
  String 
  Enter initial value 

### Get or Create a Person Group

#### Create a Scope

* Add a **scope** by clicking **+ New step**, selecting **... More** and clicking **Add a scope**Add an action 
  + New step 
  Add a condition 
  More 
  Add a switch case 
  Add a for each 
  Add a do until 
  Add a scope 
* Click on the **Ellipsis (...) menu**, select **Rename** and name this scope **Get person group scope**
* *Creating a scope just lets us group and organise Actions, and roll them up when we don't need to see them. It has no effect on Variables, in fact you currently cannot initialise a Variable with a scope (but you can set an existing one)*

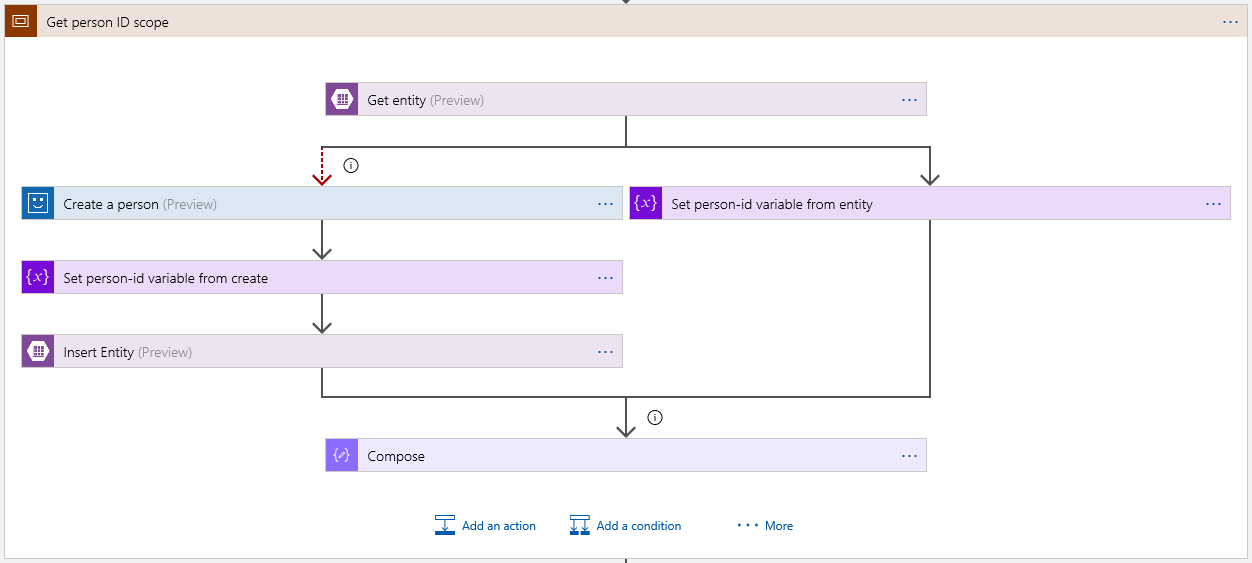
#### Try to Get an Existing Person Group

* Within the scope click **Add an action**, search for **person group** and select **Face API - Get a person group**Choose an action 
  p person g roup 
  Connectors 
  Face API 
  Triggers (O) Actions (4) 
  See more 
  See more 
  Face API 
  Face API 
  Face API 
  Face API 
  - Create a person Preview 
  - Create a person group Preview 
  - Get a person face Preview 
  - Get a person group Preview 
  TELL US WHAT YOU NEED 
  O 
  Help us decide wh'ch connectors and triggers to add next with UserVoice 
  Cancel 
* Click in the **Person Group Id** field and select **Enter custom value** from the dropdown list  
  Get a person group 
  (Preview) 
  O 
  Person Group Id 
  Connected to face-api. 
  Target Person Group Id. 
  No Items 
  Enter custom value 
* Select the **person-group-id** variable from the popup
* Your action should look like this  
  Get a person g roup 
  (Preview) 
  * Person Group Id 
  Connected to face-api. 
  person-group-id 
  Change connection. 
  x 
  Add dynamic content 
  perso n -na me 
  person-g roup-id 
  When a file is created 
* The first run through the person group won't exist, so we need to add an action which will run in the event of an error and which will create the missing person group for us.

#### Create a Person Group

* Within the scope click **Add an action**, search for **person group** and select **Face API - Create a person group**Choose an action 
  p person g roup 
  Connectors 
  Face API 
  Triggers (O) Actions (4) 
  See more 
  See more 
  Face API 
  Face API 
  Face API 
  Face API 
  - Create a person Preview 
  - Create a person group Preview 
  - Get a person face Preview 
  - Get a person group Preview 
  TELL US WHAT YOU NEED 
  O 
  Help us decide wh'ch connectors and triggers to add next with UserVoice 
  Cancel 
* Click in the **Person Group Id** field and select the **person-group-id** variable from the popup.
* Set the **Name** field to **GotG Cast**.
* Your action should look like thisCreate a person group 
  (Preview) 
  * Person Group Id 
  * Name 
  person-group-id 
  GotG Cast 
  Add dynamic content 
  Show advanced options 
  Connected to face-api. Change connection. 
  Variables 
  person-na me 
  person-g roup-id 
* You need to configure this Action to run *only* when the previous **Get a person group** Action fails; to do this click the **Ellipsis (...) menu** for the **Create a person group** action and click **Configure run after**Create a person group 
  (P review) 
  O 
  Person Group Id 
  Name 
  person-group-... 
  GotG Cast 
  Show advanced options v 
  Connected to face-api. Change connection. 
  Rename 
  Add a comment 
  Settings 
  Configure run after 
  Peek code 
  Delete 
* Untick the **is successful** checkbox and tick the **has failed** checkbox instead, the run after settings should look like this  
  'Create a person group' should run after: 
  Get a person group 
  Failed 
  Done 
  is successful 
  has failed 
  is skipped 
  has timed out 
  Cancel 
* Click **Done**
* Your action should look like this  
  Create a person group 
  (P review) 
  O 
  Person Group Id 
  Name 
  person-group-... 
  GotG Cast 
  Show advanced options v 
  Connected to face-api. Change connection. 
  Rename 
  Add a comment 
  Settings 
  Configure run after 
  Peek code 
  Delete 

### Get or Create a Person ID for a Person

* For simplicity in this lab we're going to encode the name of the person in our filenames by naming convention, e.g. **Amanda Citizen-01.jpg**
* This will be a little more complex, so here is a reference for what we’re going to build in the next few steps  
  

#### Create a scope

* Add a **scope** by clicking **+ New step**, selecting **... More** and clicking **Add a scope**
* Click on the **Ellipsis (...) menu**, select **Rename** and name this scope **Get person ID scope**
* Your scope should look like this  
  Get person ID scope 
  Add an action 
  Add a condition 

#### Try to Get an Existing Person ID from Table Storage

* Within the scope click **Add an action**, search for **Table Storage** and select the **Azure Table Storage - Get entity** action  
  Choose an action 
  able Stora e 
  Connectors 
  Azure Table 
  Storage 
  Triggers (O) Actions (12) 
  Azure Table Storage - Delete Entity Preview 
  Azure Table Storage - Get entities Preview 
  Azure Table Storage - Get entity Preview 
  Azure Table Storage - Insert Entity Preview 
  Azure Table Storage - Create table Preview 
  Azure Table Storage - Delete a table Preview 
  Azure Table Storage - Get a table Preview 
  Azure Table Stora e - Insert or Me e Enti 
  Preview 
  See more 
  See more 
  O 
  O 
  O 
  O 
  O 
  O 
  O 
  Cancel 
* Since this is the first time you're adding a Table Storage action you'll need to create a connection
* Name the connection **gibc-face-lab-tables** and select the storage account you created for this lab
* Your connection should look like this  
  Get entity 
  *Connection Name 
  * Storage Account 
  gibcgpgOI 
  gibc-face lab-tables 
  Resource Group 
  gibc-face-lab 
  Create 
  australiaeast 
  Manually enter connection information 
* Click **Create**
* For the **Table** select the **PersonIds** table you created earlier
* Click in the **Partition Key** field and select the **person-group-id** variable from the popup
* Click in the **Row Key** field and select the **person-name** variable from the popup
* Your action should look like this  
  Get entity (Preview) 
  * Table 
  * Partition Key 
  * Row Key 
  Personlds 
  person-group- 
  person-name 
  id 
  Show advanced options 
  Connected to gibc-face-lab-tables. 
  Search dynamic content 
  person-na me 
  Add dynamic content 
  person-g roup-id 
  Create a person group 
  Change connection. 
  Body 

#### Set the person-id Variable if Successful

* Within the Scope click **Add an action**, search for **Set Variable** and select the **Variables - Set variable** action  
  Choose an action 
  p Set Variablel 
  Connectors 
  {x} 
  Variables 
  Triggers (O) Actions (I) 
  Variables - Set variable 
  TELL US WHAT YOU NEED 
  See more 
  See more 
  O 
  Help us decide which connectors and triggers to add next with UserVoice 
  Cancel 
* Click in the **Name** field and select the **person-id** variable
* Click in the **Value** field, click the **Expression** tab in the popup and set the expression to  
  body('Get\_entity')['Id']
* Click on the **Ellipsis (...) menu**, select **Rename** and name this Action **Set person-id variable from entity**
* This action should only run if the **Get entity** action succeeded, so click the **Ellipsis (...) menu** again, select **Configure run after**, and ensure **has failed** is unticked (**is successful** should remain ticked)
* The run after settings should look like this  
  'Set person-id variable from entity' should run after: 
  Get entity 
  Succeeded 
  Done 
  is successful 
  has failed 
  is skipped 
  has timed out 
  Cancel 
* Click **Done**
* Your action should look like this  
  Set person-id variable from entity 
  * Name 
  * Value 
  person-id 
  Add dynamic content 
  fx body( ' Get_entity• ) Id' ] 
  Update 
  String functions 
  See more 

#### Create a Person ID if None Exists

* Now we have a person ID if the person already exists, but we need to handle the case where they don't already exist
* If there's no matching record in Table Storage the **Get entity** action will error, so we'll add a separate branch to execute in that case
* Hover over the arrow between the **Get entity** and click the **+** sign, select **Add a parellel branch**, then **Add an action**Get entity (Preview) 
  Add an action 
  Set person-id variable fro 
  Add a condition 
  * Name 
  Add a switch case 
  * Value 
  ta Add a for each 
  Add a do until 
  Add a scope 
  Add an actio Add a parallel branch 
  + New step 
  Add dynamic content 
  Add an action 
  Add a condition 
  Add a switch case 
  Add a for each 
  Add a do until 
  Add a scope 
* Search for **Create Person** and select the **Face API - Create a person** action  
  Choose an action 
  Create personl 
  Connectors 
  Capsule 
  CRM 
  Excel Online 
  (OneDrive) 
  Face API 
  Todoist 
  Workday 
  Triggers (I) Actions (13) 
  Capsule CRM - Create person Preview 
  Excel Online (OneDrive) - Create table Preview 
  Excel Online (OneDrive) - Create worksheet Preview 
  Face API - 
  Create a person 
  Preview 
  Face API - 
  Create a person group Preview 
  Todoist - Create a task 
  Todoist - Create a label 
  Todoist - Create a ro-ect 
  See more 
  See more 
  G) 
  G) 
  G) 
  G) 
  G) 
  G) 
  G) 
  G) 
  Cancel 
* Click in the **Person Group Id** field and select **Enter custom value** and select the **person-group-id** variable from the popup.
* Click in the N**ame** field and select the **person-name** variable from the popup
* This action should only run if the **Get entity** action failed, so click the **Ellipsis (...) menu** again, select **Configure run after**, and untick **is successful** and tick **has failed**
* The **Create a person** action will now run only when the **Get entity** Action fails
* The run after settings should look like this  
  'Create a person' should run after: 
  Get entity 
  Failed 
  Done 
  is successful 
  has failed 
  is skipped 
  has timed out 
  Cancel 
* Click **Done**
* Your action should look like this  
  Create a person (P review) 
  * Person Group Id 
  * Name 
  Show advanced options 
  person-group- 
  person-name 
  id 
  x 
  Add dynamic content 
  Connected to face-api. Change connection. 
  p Search dynamic content 
  person-na me 
  person-g roup-id 
  Get entity 

#### Set the person-id Variable if one was Created

* Click the **+** below the **Create a person** action and select **Add an action**, search for **Set Variable** and select the **Variables - Set Variable** action
* Click the **Name** field and select **person-id**
* Click the **Value** field and select **Person Id** under Create a person from the popup
* Click on the **Ellipsis (...) menu**, select **Rename** and name this Action **Set person-id variable from create**
* Your action should look like this  
  usea In tnls now. 
  Name 
  Value 
  Set person-id variable from create 
  person-id 
  Person Id 
  O 
  Dynamic content Expression 
  p Search dynamic content 
  Add dynamic content 
  Add an action 
  Create a person 
  Person Id 
  Create a person group 

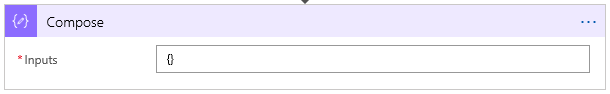
#### Save the Person ID if one was Created

* Click the **+** below the **Set person-id variable from create** action and select **Add an action**, search for **Table Storage** and select the **Azure Table Storage - Insert Entity** action  
  Choose an action 
  p Table Storage 
  Connectors 
  Azure Table 
  Storage 
  Triggers (O) Actions (12) 
  Azure Table Storage - Delete Entity Preview 
  Azure Table Storage - Get entities Preview 
  Azure Table Storage - Get entity Preview 
  Azure Table Storage - Insert Entity Preview 
  Azure Table Storage - Create table Preview 
  Azure Table Storage - Delete a table Preview 
  Azure Table Storage - Get a table Preview 
  Azure Table Storage - Insert or Merge Entity Preview 
  See more 
  See more 
  O 
  O 
  O 
  O 
  O 
  O 
  O 
  Cancel 
* Click the **Table** field and select **PersonIds** from the dropdown
* The **Entity** field is a JSON object containing the fields to be inserted into the entity, it must contain **PartitionKey** and **RowKey** properties and we will also add an **Id** field
* Copy the following empty JSON object into the **Entity** field  
  **{  
   "Id": "",  
   "PartitionKey": "",  
   "RowKey": ""  
  }**
* Place the cursor between the empty quotes **""** after **"Id"** and click **person-id** from the popup; repeat this for **PartitionKey** and **person-group-id**, and **RowKey** and **person-name**
* Your action should look like this  
  Insert Entity (Preview) 
  Table 
  Personlds 
  person-id X 
  person-group-id X 
  " Partition Key" : 
  person-name X 
  O 
  Add dynamic content 
  Add an action 
  Add dynamic content from the apps and connectors 
  used in this now. 
  Dynamic content Expression 
  p Search dynamic content 
  Variables 
  person-n am e 
  person-grou p-id 
  Hide 
  Show advanced options v 
  Connected to gibc-face-lab-tables. Change connection. 

#### Joining the Branches

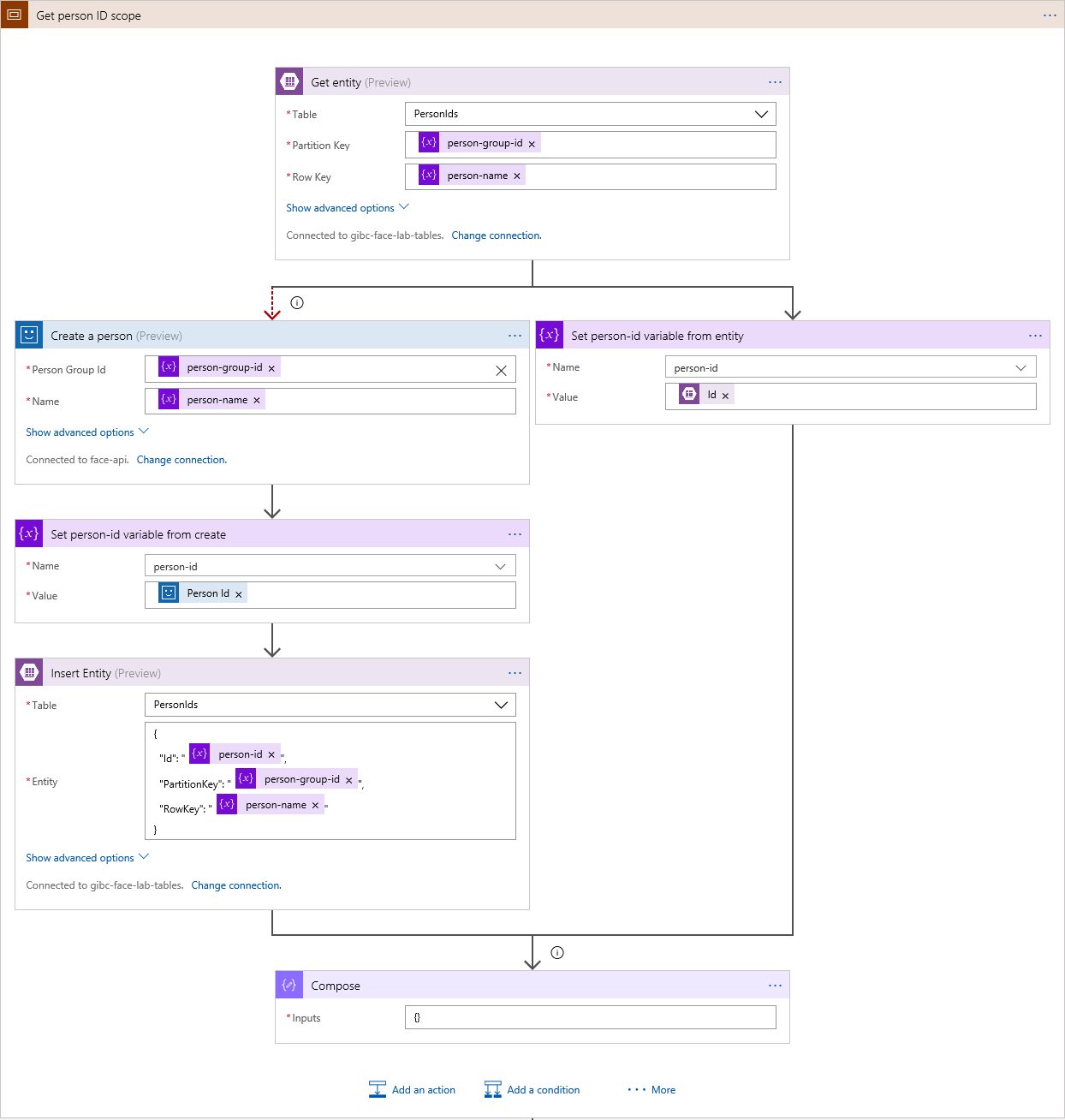
* Note that one of the branches we’ve created will always be skipped, and that means that the final state of the scope will be set to *skipped* unless we do something about it
* To fix this we'll join the branches together with an empty action which we'll set to succeed if *either* of the previous branches is successful
* At the bottom of the **scope** click the **Add an action** button, search for **Compose** and select the **Data Operations – Compose** action

Choose an action 
P Compose 
Connectors 
Data DocFusion... 
Operations 
Triggers (O) Actions (3) 
Data Operations - Compose 
DocFusion365 — SP - Compose a linked template Preview 
DocFusion365 — SP - Compose all the linked templates Preview 
TELL US WHAT YOU NEED 
See more 
See more 
O 
O 
O 
O 
Help us decide wh'ch connectors and triggers to add next with UserVoice 
Cancel 

* Click in the **Inputs** field and set the contents to an empty JSON object  
  {}
* Click the **Ellipsis (...) menu** and select **Configure run after**
* Click on the **Set person-id variable from...** tile and make sure **is successful** and **is skipped** are both ticked
* Click on the **Insert Entity** tile and make sure **is successful** and **is skipped** are both ticked
* The run after settings should look like this for both tiles  
  'Compose' should run after. 
  Set person-id variable from_. 
  {x} 
  Succeeded, Skipped 
  Insert Entity 
  x 
  Succeeded, Skipped 
  Done 
  is successful 
  has failed 
  IS skipped 
* Click **Done**
* Your action should look like this  
  

### What you Just Built

* This is the complete scope that you should have just created



## Add the Face Image to the Person

* In the Logic App click **+ New step**, search for **Add Person Face** and select the **Face API - Add a person face** action  
  Choose an action 
  p Add Person Facel 
  Connectors 
  Face API 
  Triggers (O) Actions (I) 
  Face API - Add a person face 
  Preview 
  TELL US WHAT YOU NEED 
  See more 
  See more 
  O 
  O 
  Help us decide wh'ch connectors and triggers to add next with UserVoice 
  Cancel 
* Click the **Person Group Id** field, click **Enter a custom value** and select the **person-group-id** Variable from the popup
* Click the **Person Id** field, click **Enter a custom value** and select the **person-id** Variable from the popup
* Click in the **Image Url** field. Enter the Blob endpoint you saved earlier without a trailing slash, e.g. <https://gibcgpg01.blob.core.windows.net>, and then with the cursor at the end of the URL click **Path** from the popup
* Your action should look like this  
  Add a person face 
  (Preview) 
  O 
  Add dynamic content 
  Add dynamic content from the apps and connectors 
  used in this now. 
  Dynamic content Expression 
  p Search dynamic content 
  The name of the file or folder. 
  The path of the file or folder. 
  Hide 
  Person Group Id 
  Image U rl 
  person-group-id 
  person-id X 
  https://gibcgpgOI.blob.core.windows.net 
  Path 
  Show advanced options v 
  Connected to face-api. Change connection. 

## Training the model

* *Once faces have been added, removed or changed in a Person Group we need to train the model to incorporate those changes. We train the model on an entire Person Group. For this lab we're going to train the model after each training image is added, but in the real world if we were going to make many changes we'd perform the modifications by themselves and then train the model one time after all the changes were completed. We might even train the model on a schedule, such as once a day, depending on our needs.*
* Click **+ New step**, search for **HTTP** and select the **HTTP - HTTP** action  
  Choose an action 
  Connectors 
  HTTP _ HTTp 
  H TTP 
  Azure Kusto 
  Content 
  M ode rator 
  HTTP with 
  Azure AD 
  Instagram 
  Nexmo 
  See more 
  Pitney 
  See more 
  Triggers (O) Actions (44) 
  Azure Kusto - Run control command and visualize results Preview 
  Azure Kusto - Run query and list results Preview 
  Azure Kusto - Run query and visualize results Preview 
  Content Moderator - Create Reviews for Reviewers in your moderation 
  Preview 
  team 
  Content Moderator - Detect Language of a given text input 
  content Preview 
  HTTP - HTTP + Swagger 
  HTTP - HTTP Webhook 
  O 
  O 
  O 
  O 
  O 
  O 
  O 
  Cancel 
* Set **Method** to **POST**
* Set **URL** to  
  [https://australiaeast.api.cognitive.microsoft.com/face/v1.0/persongroups//train](https://australiaeast.api.cognitive.microsoft.com/face/v1.0/persongroups/train)  
  Note the double slash before **train**, we're going to put a variable in between them.
* Place the cursor in between the **//** before **train**, and select the **person-group-id** variable from the popup
* Set the first header **key** to **Ocp-Apim-Subscription-Key** and the **value** to the **Face API Key** you saved when you created your Face API instance.
* Your action should look like this  
  Hide 
  HTTp 
  Method 
  Headers 
  Body 
  Show advanced options v 
  O 
  POST 
  https://australiaeast.api.cognitive.microsoft.com/face/vI.0/persongroups/ 
  person-group-id x .„train 
  Ocp-Apim-Subscription-Key 
  Enter key 
  Enter request content 
  Add dynamic content 
  333fdfc41fcd4e8095b74a644aa 
  b72a5 
  Enter value 
  Add dynamic content from the apps and connectors 
  used in this now. 
  Dynamic content Expression 
  p Search dynamic content 
  person-n am e 
  person-grou p-id 
  Add a person face 

### Try it Out

* Drop one or more training photos named in this format [Person Name]-[Number].jpg (e.g. Amanda Citizen-01.jpg) into the training folder that you selected for the OneDrive trigger
* After a few seconds the image should be picked up, a person ID retrieved or created, and the person group trained
* You can use the Logic Apps **Overview** to see the status of runs
* You can now go back to your identify-faces Logic App and change the Face API key to your own key, and try recognising faces for people that you’ve trained a model on
* *NOTE: If you drop multiple image files into the folder at the same time you may see some runs marked as failed. This is because you can’t start training the model while another training session is already running. In reality you would create a mechanism so that this didn’t occur, but for the purposes of simplicity this has been omitted for this lab.*