Download https://github.com/szelor/Hekaton repository to your local machine and unpack it to c:\azureml folder.

# Lab Cognitive Services

## Try out the Cognitive Services using the website demo options

Navigate to: https://azure.microsoft.com/en-gb/services/cognitive-services/directory/vision/. There are lots of different demos to try in each section (Scene and Activity Recognition in Images, OCR, Face Detection, Emotion Detection, Video indexer etc).

Select the Demo link next to Scene and activity recognition in images under Computer Vision. There are also other demo links to explore the different services.

Now select Browse button and upload the cat.jpeg or city.jpeg image from sample-images/computer-vision-web-browser/.

Set up your Azure Account

You may activate an Azure free trial at https://azure.microsoft.com/en-us/free/.

If you have been given an Azure Pass to complete this lab, you may go to http://www.microsoftazurepass.com/ to activate it. Please follow the instructions at https://www.microsoftazurepass.com/howto, which document the activation process. A Microsoft account may have one free trial on Azure and one Azure Pass associated with it, so if you have already activated an Azure Pass on your Microsoft account, you will need to use the free trial or use another Microsoft account.

## Provision Cognitive Services keys

First log into Microsoft Azure and choose Portal in the top right corner. Once in the portal select Create a resource and search Cognitive Services and choose Enter. Then select Create on the Cognitive Services blade.

Enter details to create an account:

* Name: enter a suitable name for the service (example: hekatoncognitive)
* Subscription: Choose your subscription
* Location: Choose West US
* Pricing Tier: S0
* Resource Group: Select 'Create new', and provide a sensible name (example hekaton)
* select the checkbox after reading the terms below
* select 'Create'

Once created, in your notifications (top right corner) select go to resource.

In the Cognitive Services page, select Keys and copy KEY 1. Save it in a text file so you can easily access it in future labs.

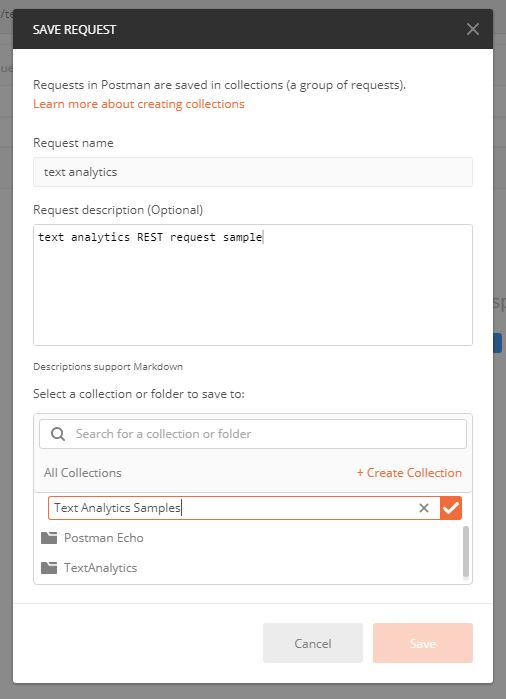
Now select Overview in the left hand pane and copy the Endpoint variable.

## Text Analytics via REST

Download (https://www.getpostman.com/downloads/) and open Postman, an API Development environment on your local machine.

Select Request

Enter request details as below and choose the option create a new collection and name it "Text Analytics Samples"



Select the newly created collection and choose save

Now create a request to call your text analytics API:

* Change from a GET request to a POST request in the top left
* Enter your endpoint URL and add text/analytics/v2.0/sentiment to the end
* Select Headers underneath the URL box
* In Key type Ocp-Apim-Subscription-Key and in Value add your KEY1 value
* In Key type Content-Type and in Value type application/json
* Select Body underneath the URL box
* Select raw from the radio button options
* Copy JSON sample from sample-code/cognitive-services-api-task/sentiment-analysis-text.json into the box
* Select the Send button and review the Response

You can also try other options from the REST API - such as KeyPhrases function. Change the end of the URL from sentiment to keyPhrases and select send to view the key phrases for the example text.

Check out the language support for the Text Analytics API. If your language is supported please edit the JSON file to translate the text and show the functionality of the API above. There is an example of a French JSON file in sample-code/text-analytics-demo/sentiment-analysis-text-fr.json.

# Lab Custom Vision and Logic Apps

Using Microsoft Azure Custom Vision service you can start to build your own personalized image classification and object detection algorithms with very little code. In this exercise we will create a dog-breed classification algorithm using Dog images from the ImageNet open dataset created by Standford University.

We have 7 Classes of dogs each with 30 images available in \Cognitive Services\sample-images\dogs folder

* Beagle
* Bernese Mountain Dog
* Chihuahua
* Eskimo Dog (aka Husky)
* German Shepherd
* Golden Retriever
* Maltese

There is also a set of test images (not for training) in this folder.

## Resource provisioning

First create a Custom Vision instance in your Azure account.

* Go to the Azure Portal main dashboard.
* Click 'Create a Resource' in the top left
* Search for 'Custom Vision'
* On the description pane for Custom Vision click Create.
* Enter details to create
  + a name for the service
  + select your subscription
  + Please choose West US as the data centre
  + Choose the S0 tier for both 'Prediction pricing tier' and Training pricing tier
  + select the resource group you created previously for this project (e.g. ainights)
  + Click Create

## Create Custom Vision project

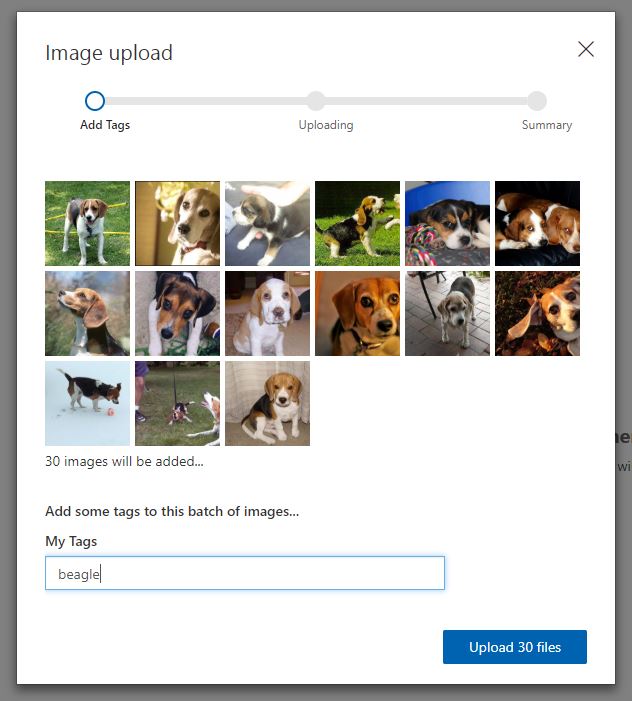
Now we can build our classifier, navigate to https://www.customvision.ai and choose sign in. Sign in with your Azure credentials account.

Once loaded choose 'New Project' which opens a window to enter details

* Name: choose a suitable name
* Description: add a description of the classifier (example shown in image below)
* Resource Group: choose the resource group you created your custom vision service in (example: hekaton[SO])
* Project Types: Classification
* Classification Types: Multiclass (Single tag per image)
* Domains: General

Choose 'Create Project' and you will land on an empty workspace. Now we can start adding images and assigning them tags to create our image classifier.

In the top left, select 'Add images', browse for the first folder of images from the Dogs folder - Beagle - and select all 30 of the images in the folder. Add the tag 'beagle' to the Beagle dog images and select 'Upload 30 files'.



Once successful you receive a confirmation message and you should see your images are now available in the workspace.

Now complete the same steps of uploading and tagging images for the other 6 dog categories in the folder. For each type of dog:

* Click add images
* Select the 30 new dog images
* Add the class label (beagle, german-shepherd, maltese etc)
* choose upload
* confirm images uploaded into the workspace

Now you should have all categories uploaded and on the left hand side you can see your dog classes and you can filter depending on type of dog image.

Now you are ready to train your algorithm on the dog image data you have uploaded. Select the green 'Train' button in the top right corner.

Once the training process is complete it will take you to the Performance tab. Here you will receive machine learning evaluation metrics for your model.

Now we have a model we need to test the model. Choose the 'Quick Test' button in the top right (next to the train button) this will open a window where you can browse for a local image or enter a web URL.

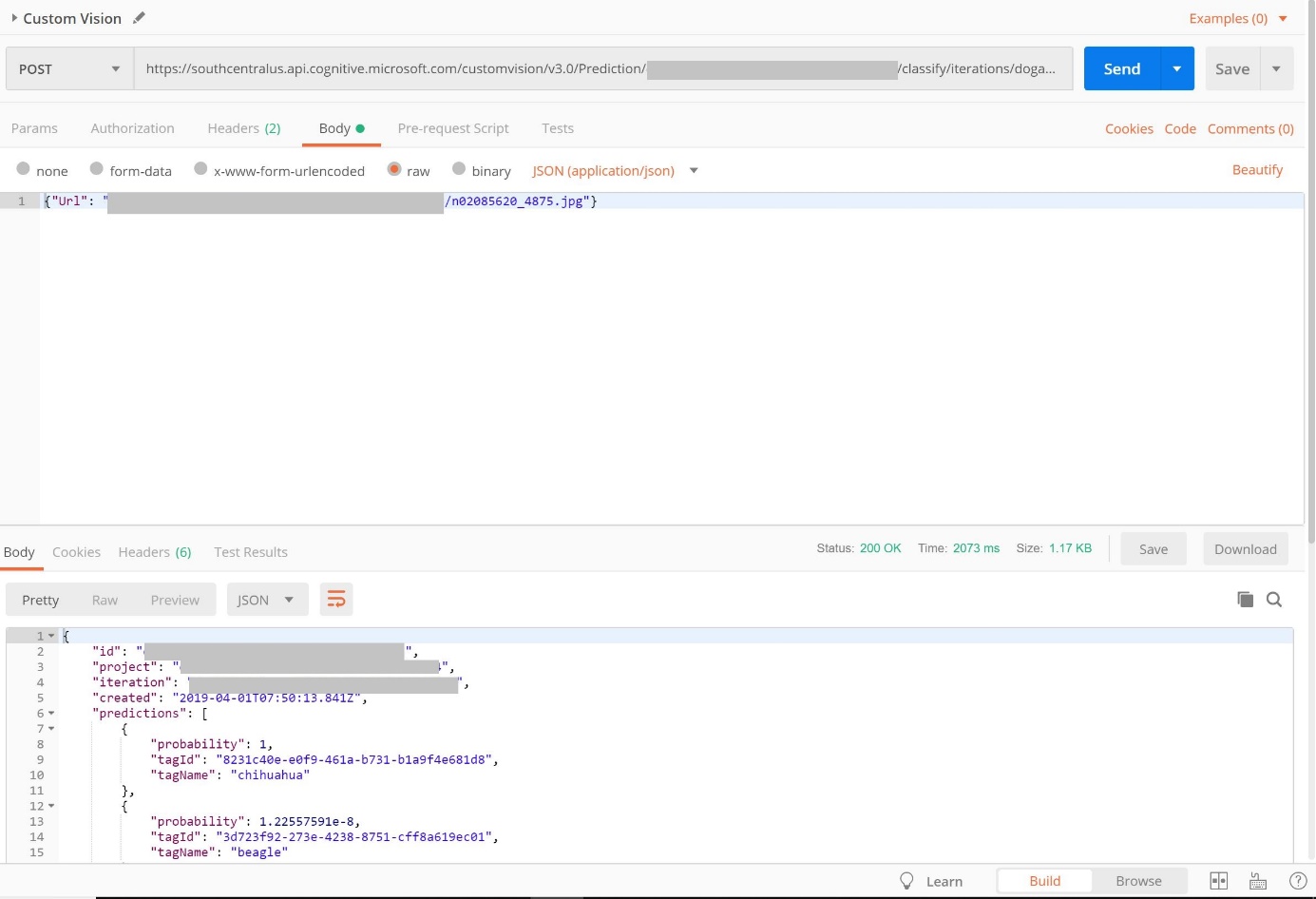
Browse for an image in the test folder (images the model have not been trained on) and upload. The image will be analysed and a result returned of what dog the model thinks it is (prediction tag) and the models confidence of its result (prediction probability).

If you click on the 'Predictions' tab on the top toolbar - you should see all the test images you have submitted. This section is for re-training, as you get new data you can add this to your model to improve its performance. The images are ordered by importance - the image, which if classified correctly, will add the most new information to the model is listed first. Whereas the last image might be very similar to other images already learnt by the model so this is less important to classify correctly. To add these images to the model - select the first image, review the results the model provided and then in the 'My Tags' box enter the correct tag and click 'save and close'.

This image will disappear from the your predictions workspace and be added to the training images workspace. Once you add a few new images and tags you can re-train the model to see if there are improvements.

To use this model within applications you need the prediction details. Therefore, you have to go to the Performance tab from the top bar, click the Publish button and provide a name for this published iteration.

You can now select the Prediction URL button to gain all information you need to create a Postman call to your API, by setting the URL, the Header and the Body (using both an image or an image URL)



Great work! you have created your specialised dog classification model using the Azure Custom Vision Service

## Build Custom AI into an Application using Azure Logic Apps

In this section you will build an Azure Logic App to consume your Custom Vision AI dog classification application

First we need to create two Azure Storage Accounts.

Go to the azure portal and click create new resource in the top left corner. Select the section Storage and choose the first option Storage Account. We are going to create two storage accounts:

* one for the images to be dropped into to be processed (called storagea)
* another for the results after processing to be uploaded to (called storageb)

Complete the process below twice so you have two storage accounts in total

On the storage account creation page enter options to setup your storage account:s

* Subscription: choose your subscription
* Resource Group: choose the resource group you have been using for this workshop (e.g. ainights)
* Storage Account Name: (must be unique) enter an all lowercase storage account name. Such as ainightsstor(yourname) or resultsainights(yourname) - append your name to the end of the storage account name so you know its unique (remove the brackets)
* Location: your closest data center
* Performance: Standard
* Account Kind: Blob Storage
* Replication: Locally-redundant storage (LRS)
* Access Tier: Hot
* Select Review + create, confirm validation is passed and then select Create.

Once your deployment is complete, got to the resource and review the account settings. Select Blobs to review your empty blob storage account.

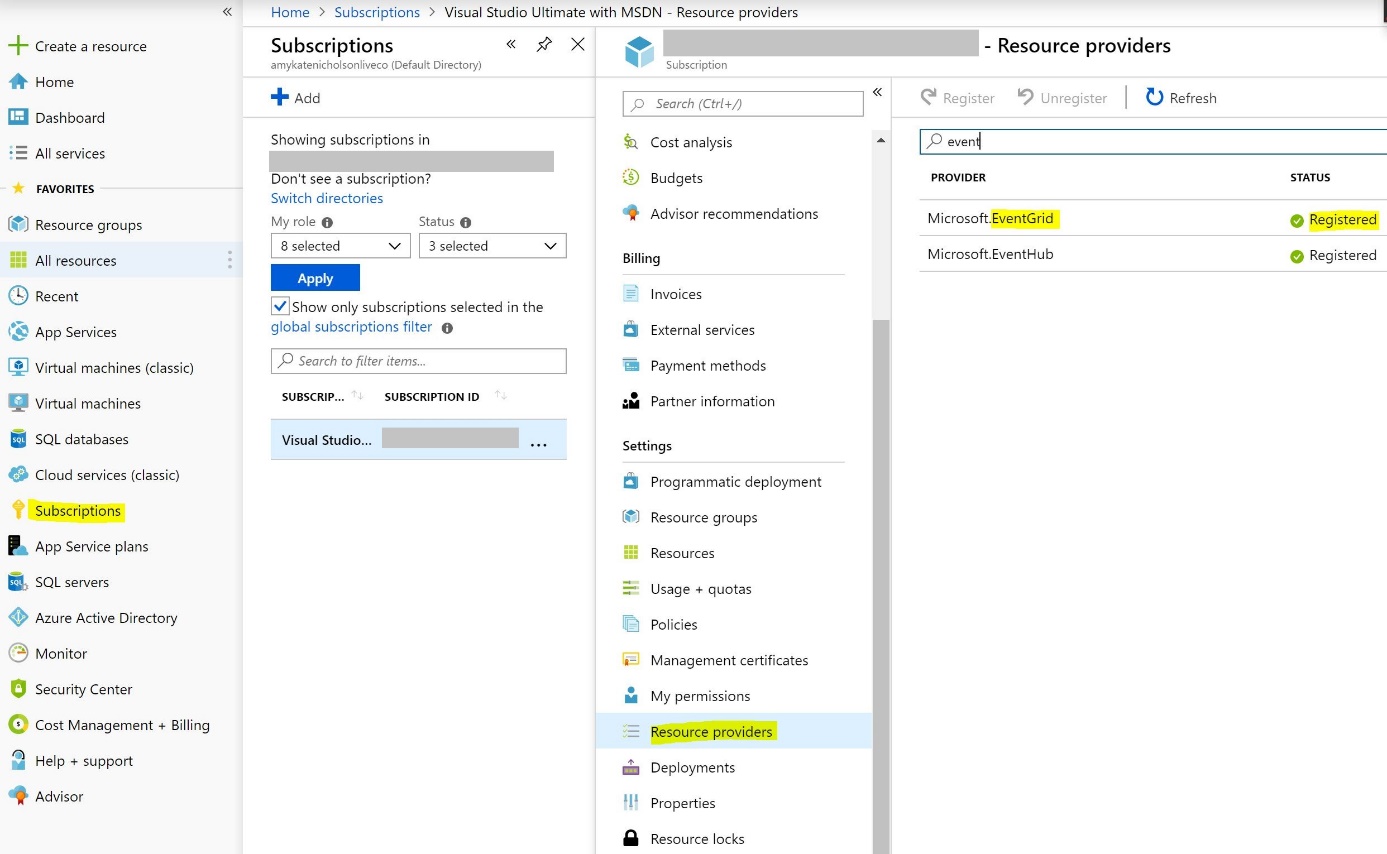
We need to add a container to the storage account to store our images and results.

* Select the + Container button and create a name for the container
* an example for the ainightsstor account would be images
* an example for the resultsainights account would be results
* For the public access level setting select Container (anonymous read access for containers and blobs)

Now we will create a Logic App - this will connect your image storage account to your AI classification service and put the results in your results storage account.

Head to the Azure Portal Homepage. We are going to use Event Grid, a service that detects triggers in an Azure subscription (in our case, when a new blob is created in your Azure Storage account). Before we build with this - we must register it.

Got to subscriptions in the left panel, select your subscription and find Resource Providers in the left pane. Once the resource providers are listed - search "event" and select Microsoft.EventGrid.



If this is not already status registered, select register from the toolbar

Once registered with a green tick - go back to the Azure Portal Homepage. Select Create a Resource. Type Logic App and select the service

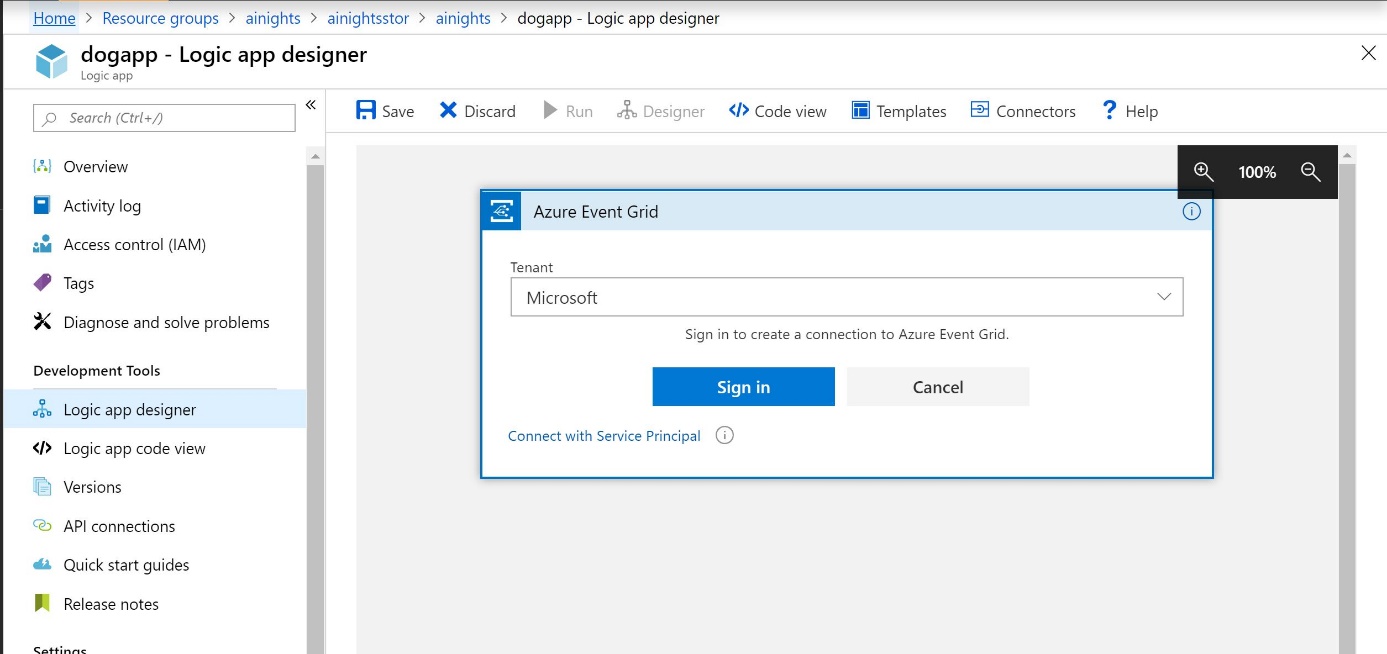
Create the logic app by entering some setup detail like below:

* Name: suitable name for the dog classification application
* Subscription: Choose your subscription
* Resource Group: (use existing e.g. ainights) select the resource group you have been working for the whole workshop
* Location: choose the data center closest to you
* Log Analytics: off

Choose Create

Once created, go to resource. From here we can create our logic process. Select Logic app designer from the left menu and then the When an Event Grid resource event occurs option.

Connect to Azure event grid by signing in using your Azure credentials



Once connected and you see a green tick, select continue.

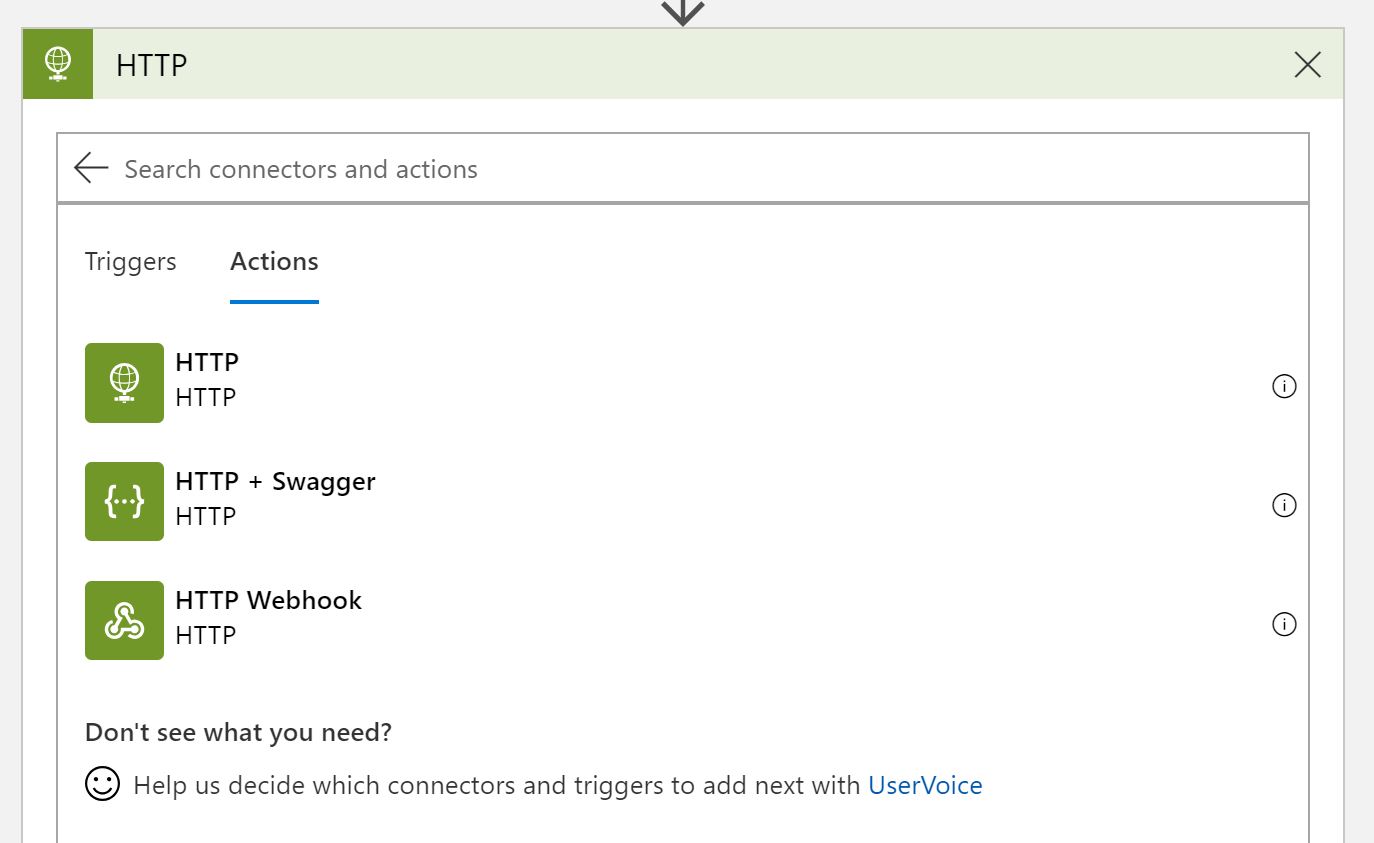
Select the options below:

* Subscription: your subscription
* Resource Type: Microsoft.Storage.StorageAccounts
* Resource Name: choose your image storage account (e.g. ainightsstor)
* Event Type Item - 1: Microsoft.Storage.BlobCreated

Then choose next step. Type Parse JSON and select the parse JSON operator as part of the data Data Operations category

* Content: select the box and from the Dynamic Content box on the right, select Body
* Schema: select this box and enter the JSON schema provided in the logic-app-schema1 file

Then choose next step. Type HTTP and select the HTTP option as below



Now we need to fill in the details of the REST API request - similar to using Postman App.

* Method: POST
* URI: enter Prediction URL from Custom Vision Service
* Headers:
  + "Prediction-Key" : enter your prediction key from the custom vision service
  + "Content-Type" : "application/json"
* Queries: enter nothing
* Body: {"Url": "REPLACE WITH DYNAMIC CONTENT URL"}

Then choose next step. Type Parse JSON and select the parse JSON operator again as part of the data Data Operations category

* Content: select the box and from the Dynamic Content box on the right, select Body
* Schema: select this box and enter the JSON schema provided in the logic-app-schema2 file

Choose next step.

Type for each and select the grey control step called for each Once selected in the output from previous step box, select the box and from Dynamic content select predictions from the Parse JSON 2 category.

Choose Add an action.

Search Control, select the control icon and then from the results, select Condition

In the If True box select Add an action

Search for Azure Blob Storage and select the icon for Create Blob

In connection name enter results and select your results blob storage account name from the listed options and select create

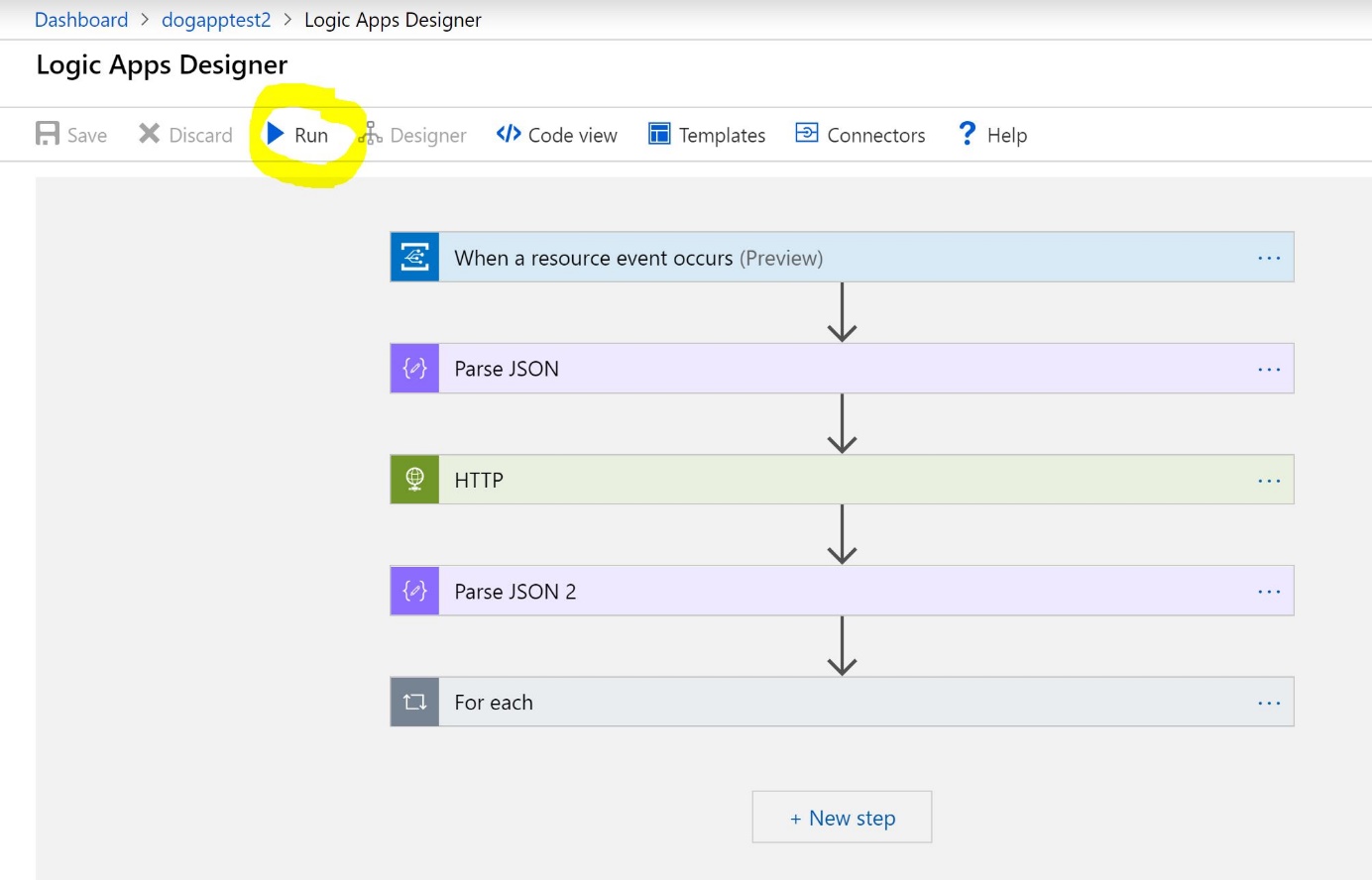
In folder path, select the folder icon, far right, and choose the container name you created that is populated

Select the Blob name field and enter: result-(then from the Dynamic content box under Parse Json (1) select id)

Under Blob Content, select the field and in the Dynamic Content box on the right, select see more under the Parse Json 2 section. Then select tagName, enter a colon ":" and then select probability.

Finally save the logic app in the top action bar

Once saved, lets test the app for the desired outcome. Select Run from the top action bar



Now navigate to your images storage account (easy to find from the resource group section). Choose Blob and select the images container. In there you should see an upload button. Upload one of the images from the Dogs data testset folder.

Once uploaded, navigate back to your Logic App main page and review the runs history. All sections should have a green tick and you can select each one to view the input and output between the layers (this is also a great way to debug if it doesn't run as expected).

Finally navigate to your results blob storage account, select blob, enter the results container and review the file now created there.