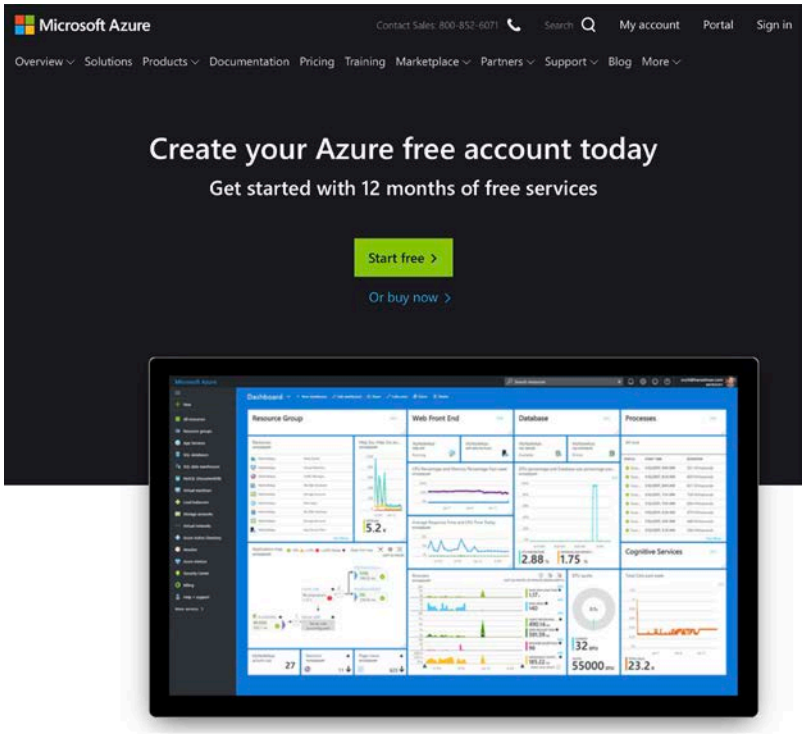
Activity 1 – Training an Image Classifier

In this activity, we will:

- ❑ learn how to build a classifier through the Microsoft's Custom Vision Service.
- ❑ Use a sample python script to use your classifier.

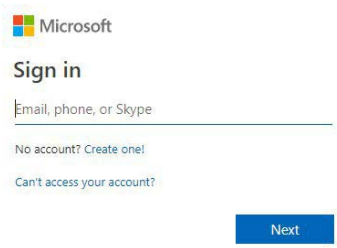
1. Prerequisites

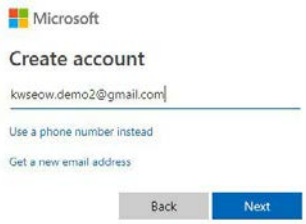
a) A valid Azure subscription. Create an account for free (https://azure.microsoft.com/en-us/free/).



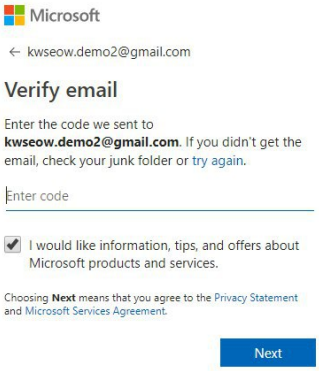
b) Click on "**Start Free**"

c) Click on **Create One** if you do not have an existing account.

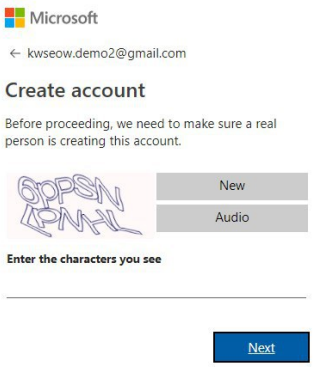




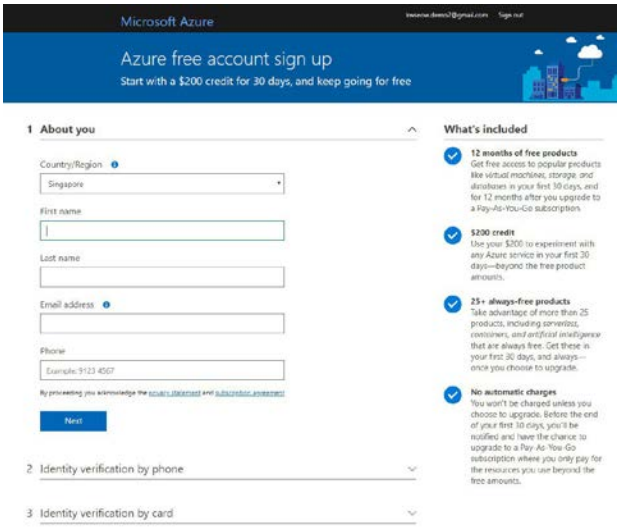
d) You will receive a verification email. Use the code provided in the email to version your account.

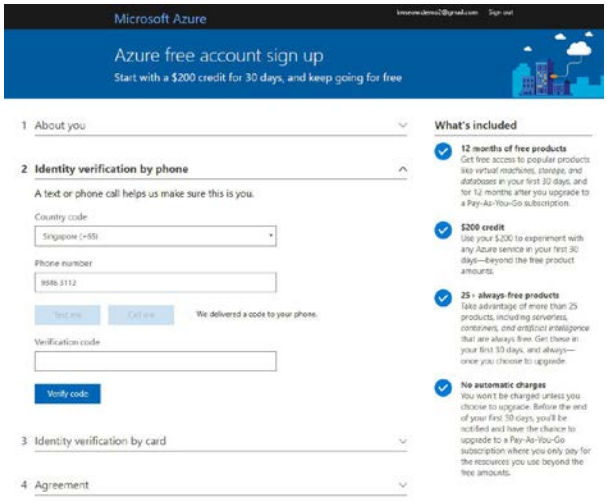


e) You will be asked to do a visual verification by entering some prompted text.



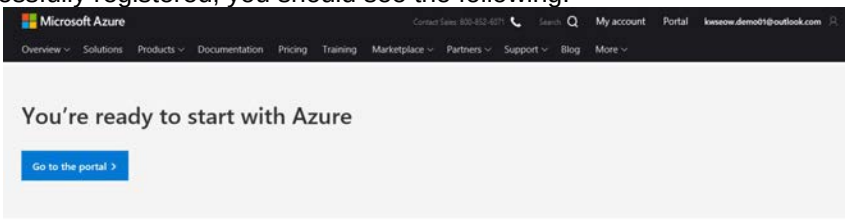
f) If entered correctly, you be presented with a screen to enter your Azure account information for verification.



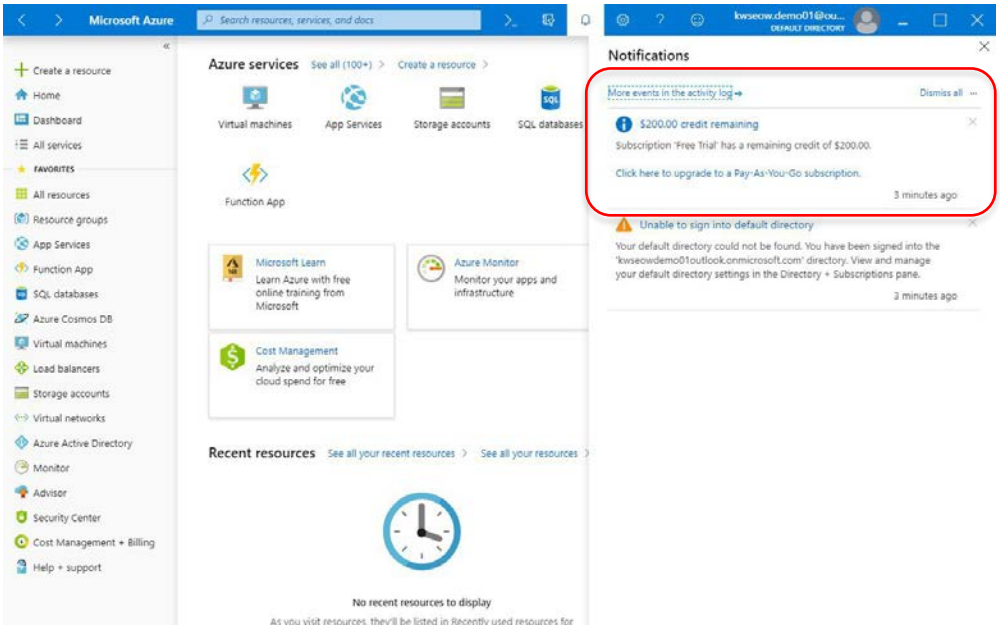


- You will be asked to enter your credit card information for verification too.
- *You may need to use a different mobile number, credit card and use outlook.com*

g) Once you are successfully registered, you should see the following.



h) Check if you received the $200 credit.



2. A set of images

To start training your own classifier, you will need a set of images.

As a minimum, we recommend you use at least 30 images per tag in the initial training set. You'll also want to collect a few extra images to test your model once it is trained.

In order to train your model effectively, use images with visual variety. Select images with that vary by:

- camera angle
- lighting
- background
- visual style
- individual/grouped subject(s)
- size
- type

Additionally, make sure all of your training images meet the following criteria:

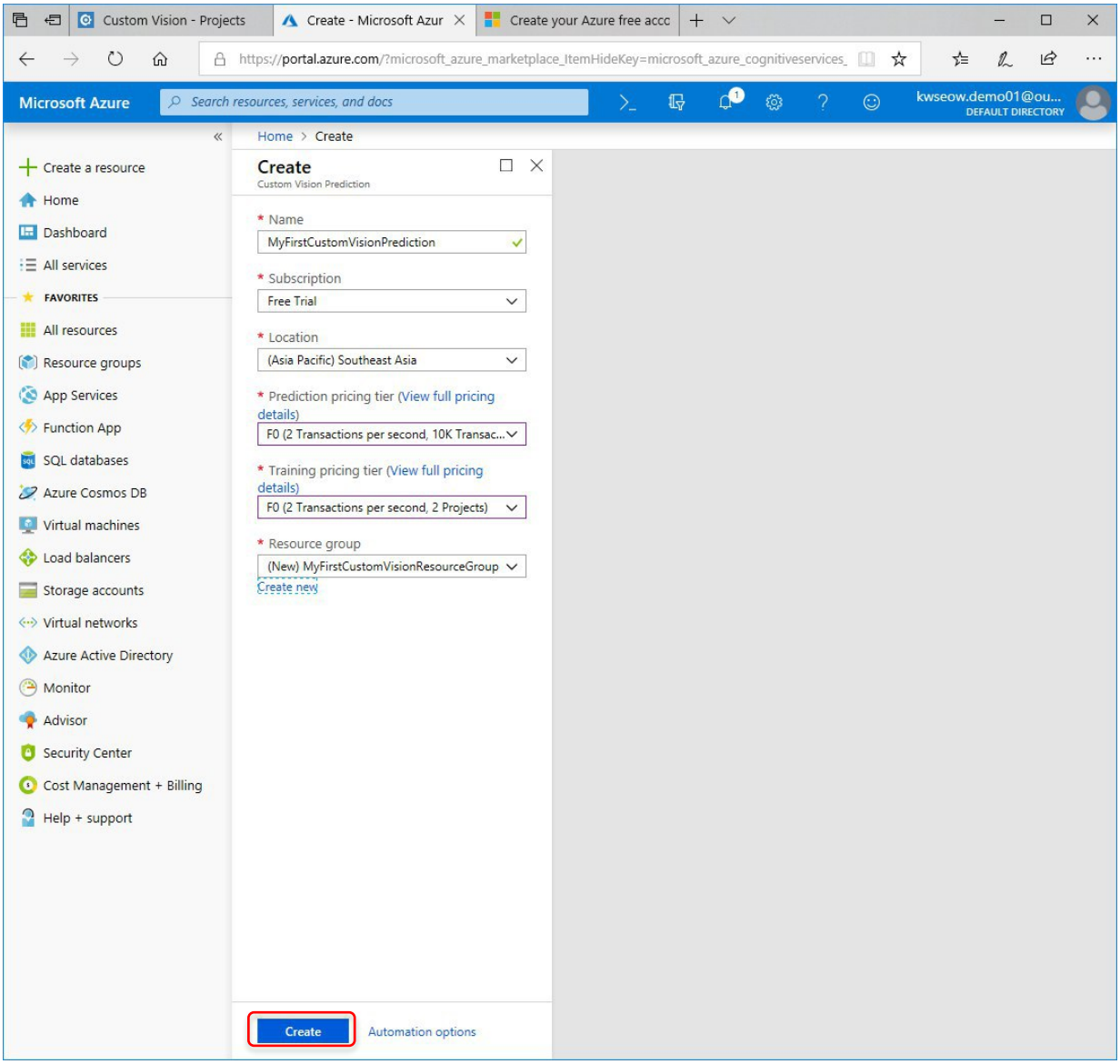
1) .jpg, .png, or .bmp format
2) no greater than 6MB in size (4MB for prediction images)
3) no less than 256 pixels on the shortest edge; any images shorter than this will be automatically scaled up by the Custom Vision Service

3. Create Custom Vision resources in the Azure portal *use Microsoft Edge or Chrome

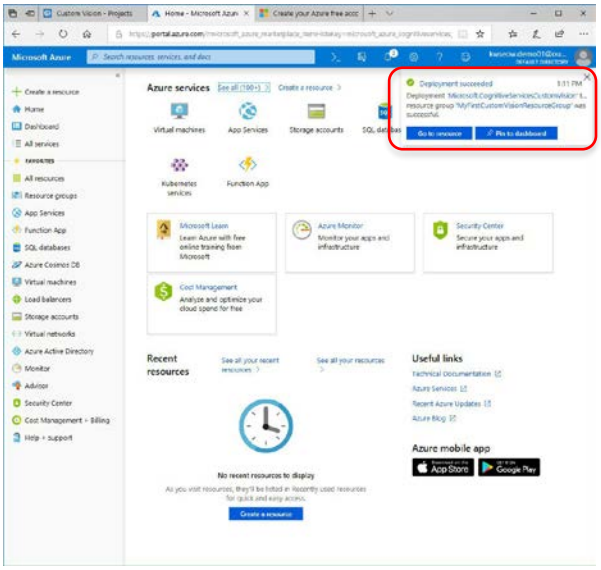
To use Custom Vision Service, you will need to create ***Custom Vision Training and Prediction*** resources in the in the Azure portal. This will create both a Training and Prediction resource.

a. Create Custom Vision resources in Azure portal (https://portal.azure.com/?microsoft_azure_marketplace_ItemHideKey=microsoft_azure_cognitiveservices_customvision#create/Microsoft.CognitiveServicesCustomVision)

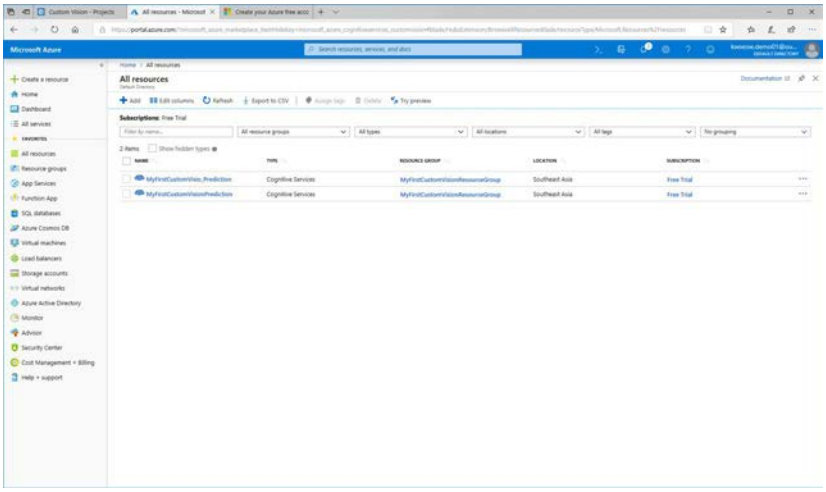
Enter the details as shown below to create the resource. * *Create a new Resource Group if necessary*.



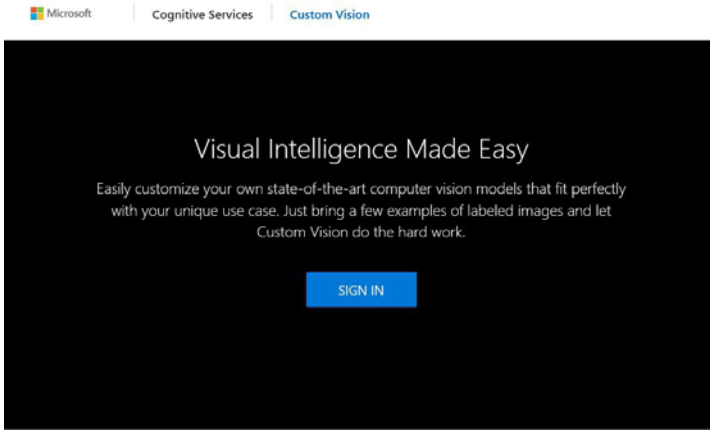
It may take a few minutes to provision and deploy the resources. You may see notifications on the top right corner of the browser.



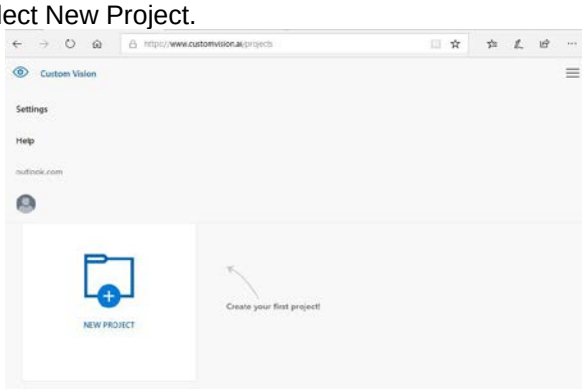
You can select **All resources** on the left pane, **Favorites**, to see the newly created resources



b. Create a new Project
In your web browser, navigate to the Custom Vision web page (https://customvision.ai/) and select Sign in. Sign in with the same account you used to sign into the Azure portal.



c. To create your first project, select New Project.



* please confirm you have selected the same “Directory” in the Custom Vision portal as the directory in the Azure portal where your Custom Vision resources are located. In both sites, you may select your directory from the drop down account menu at the top right corner of the screen.

d. The Create new project dialog box will appear.

a) Enter a **name** and a **description** for the project.

b) Select a **Resource Group**. If your signed-in account is associated with an Azure account, the Resource Group dropdown will display all your Azure Resource Groups that include a Custom Vision Service Resource.

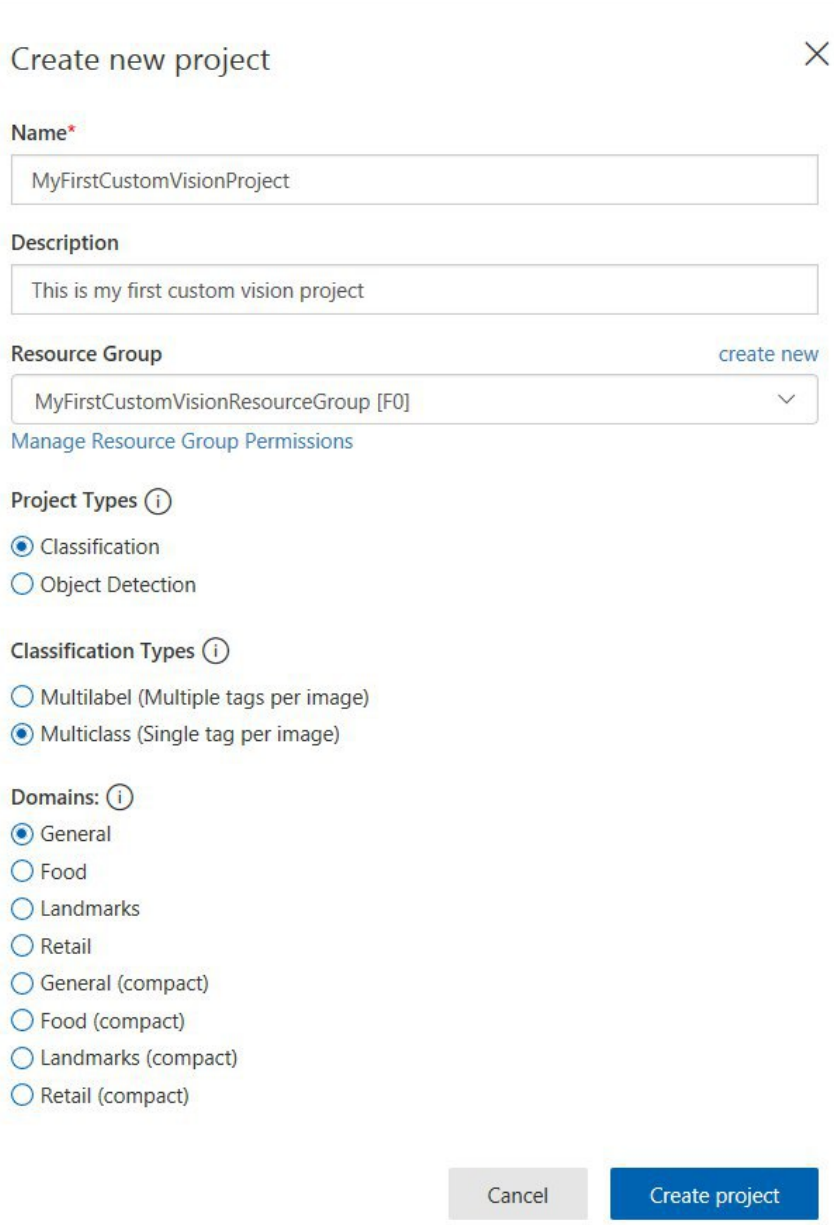
c) Select **Classification** under Project Types. Then under Classification Types, choose either Multilabel or Multiclass, depending on your use case. Multilabel classification applies any number of your tags to an image (zero or more), while multiclass classification sorts images into single categories (every image you submit will be sorted into the most likely tag). You will be able to change the classification type later if you wish.

d) Next select one of the available **domains**. Each domain optimizes the classifier for specific types of images,

as described in the following tables. You will be able to change the domain later if you wish.

Domain	Purpose
Generic	Optimized for a broad range of image classification tasks. If none of the other domains are appropriate, or you are unsure of which domain to choose, select the Generic domain.
Food	Optimized for photographs of dishes as you would see them on a restaurant menu. If you want to classify photographs of individual fruits or vegetables, use the Food domain.
Landmarks	Optimized for recognizable landmarks, both natural and artificial. This domain works best when the landmark is clearly visible in the photograph. This domain works even if the landmark is slightly obstructed by people in front of it.
Retail	Optimized for images that are found in a shopping catalog or shopping website. If you want high precision classifying between dresses, pants, and shirts, use this domain.
Compact domains	Optimized for the constraints of real-time classification on mobile devices. The models generated by compact domains can be exported to run locally.

e) Finally, select Create project



4. Choose training images

As a minimum, we recommend you use at least 30 images per tag in the initial training set. You'll also want to collect a few extra images to test your model once it is trained.

In order to train your model effectively, use images with visual variety. Select images with that vary by:

- camera angle
- lighting
- background
- visual style
- individual/grouped subject(s)
- size
- type

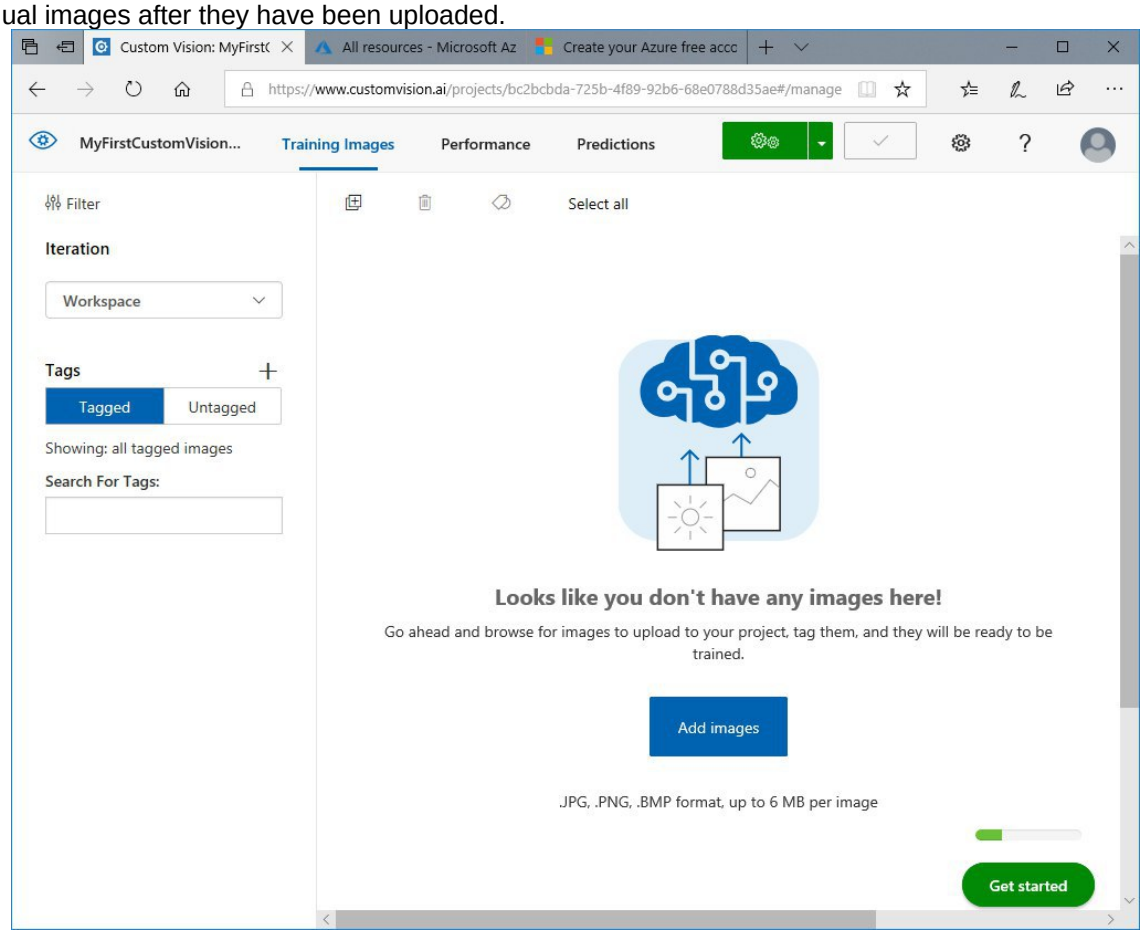
Additionally, make sure all of your training images meet the following criteria:

4) .jpg, .png, or .bmp format
5) no greater than 6MB in size (4MB for prediction images)
6) no less than 256 pixels on the shortest edge; any images shorter than this will be automatically scaled up by the Custom Vision Service

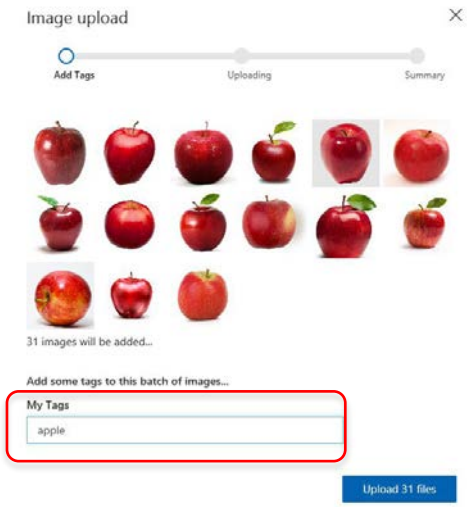
a) Download fruits images from here: http://bit.ly/2RCLYEV
b) Unzip the file and you should see images of fruits in various sub directories.

5. Upload and tag images

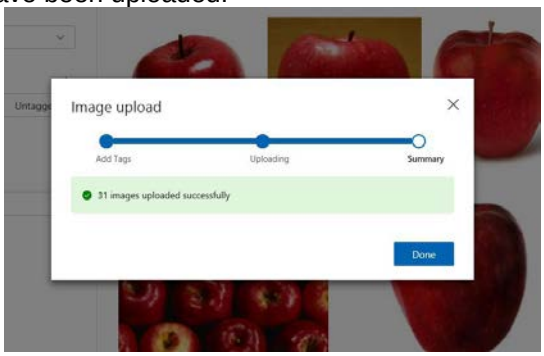
a) To add images, click the **Add images** button and then select Browse local files. Select **Open** to move to tagging. Your tag selection will be applied to the entire group of images you've selected to upload, so it is easier to upload images in separate groups according to their desired tags. You can also change the tags for individual images after they have been uploaded.



b) To create a tag, enter text in the **My Tags** field and press Enter. If the tag already exists, it will appear in a dropdown menu. In a multilabel project, you can add more than one tag to your images, but in a multiclass project you can add only one. To finish uploading the images, use the **Upload [number]** files button



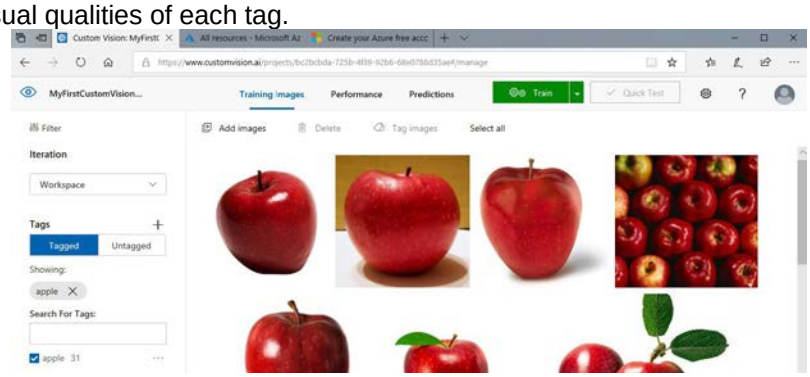
c) Select Done once the images have been uploaded.



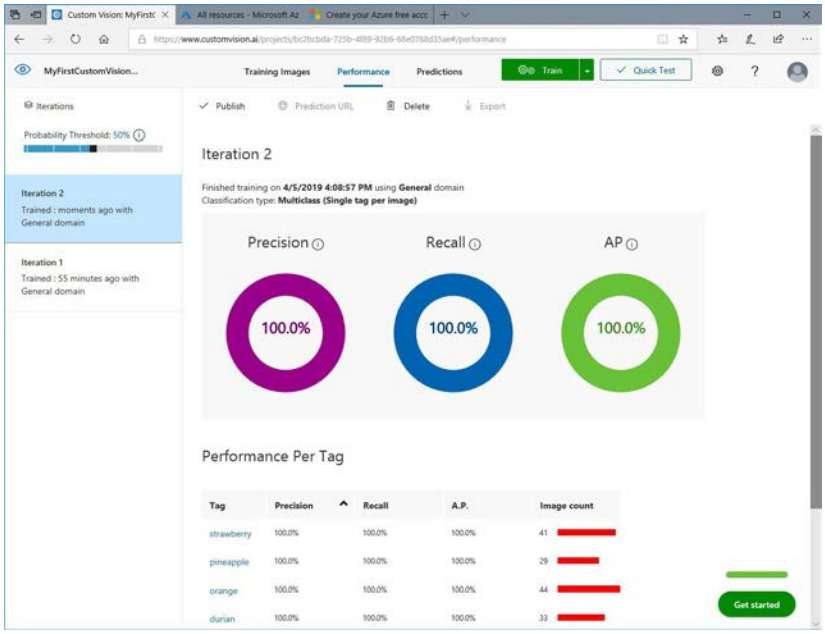
d) To upload another set of images, return to the top of this section and repeat the steps.
e) Repeat the steps to upload images of bananas, durians, strawberries etc

6. Train the Classifier

a) To train the classifier, select the **Train** button. The classifier uses all of the current images to create a model that identifies the visual qualities of each tag.



b) The training process should only take a few minutes. During this time, information about the training process is displayed in the **Performance** tab.



7. Evaluate the Classifier

After training has completed, the model's performance is estimated and displayed. The Custom Vision Service uses the images that you submitted for training to calculate precision and recall, using a process called k-fold cross validation. Precision and recall are two different measurements of the effectiveness of a classifier:

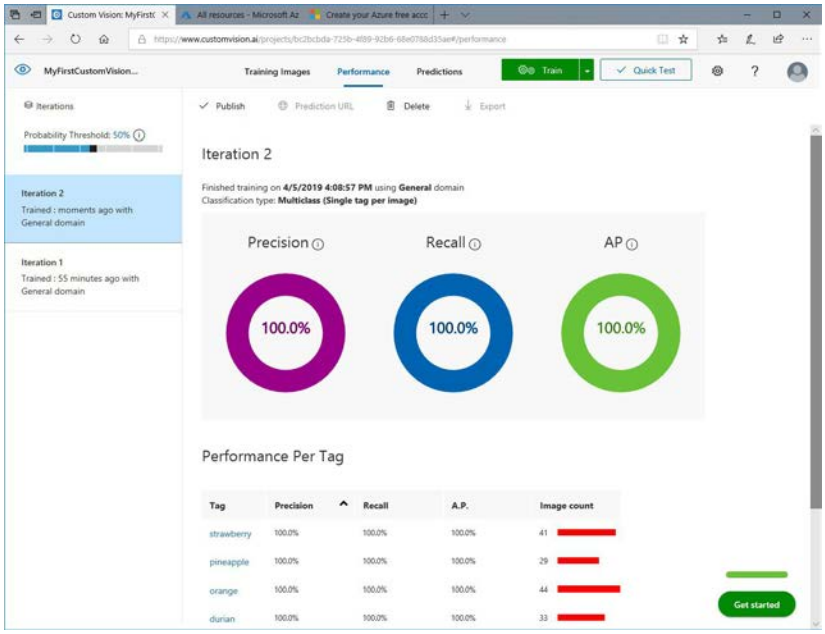
- **Precision** indicates the fraction of identified classifications that were correct. For example, if the model identified 100 images as dogs, and 99 of them were actually of dogs, then the precision would be 99%.
- **Recall** indicates the fraction of actual classifications that were correctly identified. For example, if there were actually 100 images of apples, and the model identified 80 as apples, the recall would be 80%.
- **Probability Threshold** - Note the Probability Threshold slider on the left pane of the Performance tab. This is the threshold for a predicted probability to be considered correct when computing precision and recall. Interpreting prediction calls with a high probability threshold tends to return results with high precision at the expense of recall (the found classifications are correct, but many were not found); a low probability threshold does the opposite (most of the actual classifications were found, but there are false positives within that set). With this in mind, you should set the probability threshold according to the specific needs of your project. Later, on the client side, you should use the same probability threshold value as a filter when receiving prediction results from the model.

Manage training iterations

Each time you train your classifier, you create a new iteration with its own updated performance metrics. You can view all of your iterations in the left pane of the Performance tab. In the left pane you will also find the Delete button, which you can use to delete an iteration if it's obsolete. When you delete an iteration, you delete any images that are uniquely associated with it.

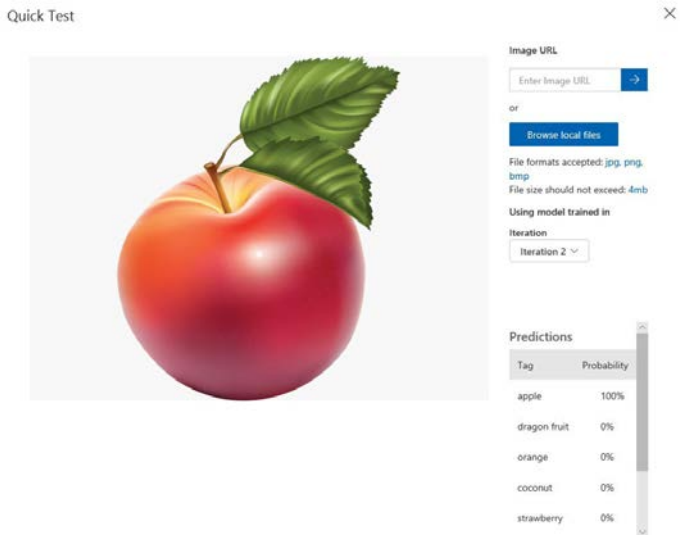
8. Test your model

a) From the Custom Vision web page, select your project. Select Quick Test on the right of the top menu bar. This action opens a window labelled Quick Test.



b) In the **Quick Test** window, click in the **Submit Image** field and enter the URL of the image you want to use for your test. If you want to use a locally stored image instead, click the **Browse local files** button and select a local image file.

The image you select appears in the middle of the page. Then the results appear below the image in the form of a table with two columns, labelled Tags and Confidence. After you view the results, you may close the Quick Test window.

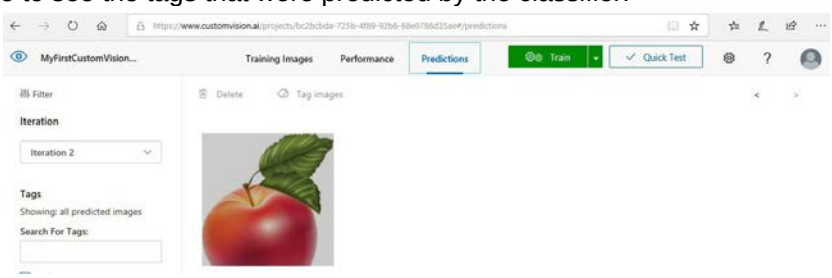


c) You can now add this test image to your model and then retrain your model

9. Use the predicted image for training model

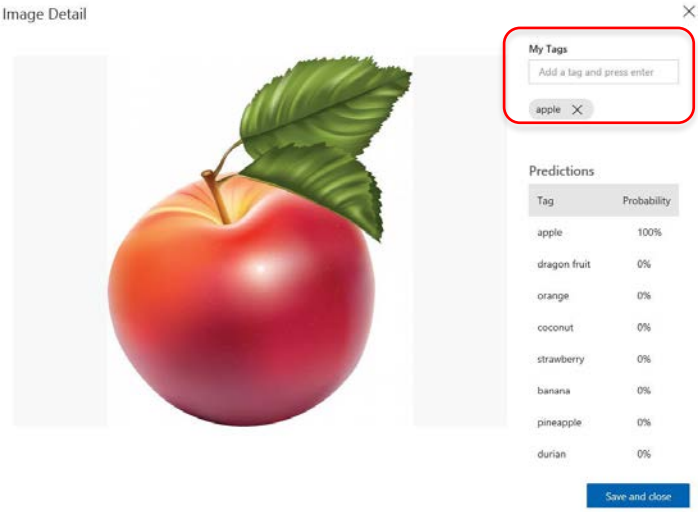
a) To view images submitted to the classifier, open the Custom Vision web page and select the **Predictions** tab.

b) Hover over an image to see the tags that were predicted by the classifier.



c) To add an image to your training data, select the image, select the tag, and then select Save and close. The image is removed from Predictions and added to the training images. You can view it by selecting the Training

Images tab.



d) Use the Train button to retrain the classifier.

10. How to improve your classifier

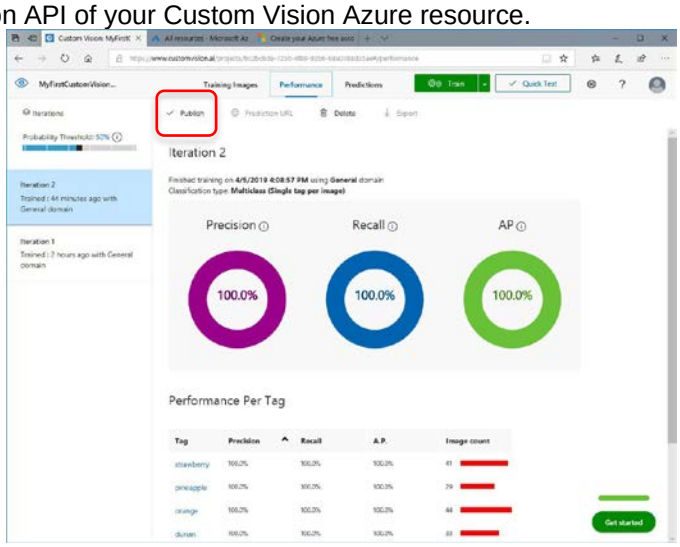
https://docs.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/getting-started-improving-your-classifier

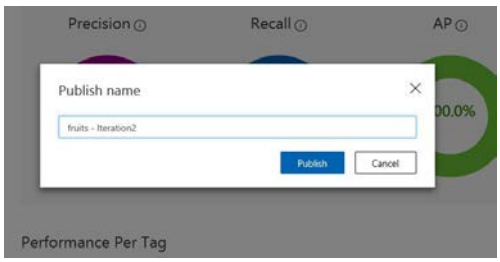
11. Using the prediction model using Python (demo)

After you've train your model, you can test images programmatically by submitting them to the Prediction API endpoint.

a) **Publish your trained iteration**. From the Custom Vision web page, select your project and then select the **Performance** tab.

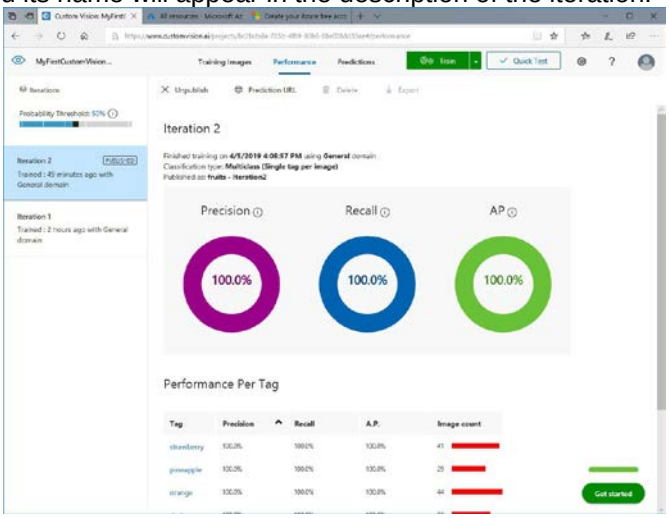
To submit images to the Prediction API, you will first need to publish your iteration for prediction, which can be done by selecting **Publish** and specifying a name for the published iteration. This will make your model accessible to the Prediction API of your Custom Vision Azure resource.



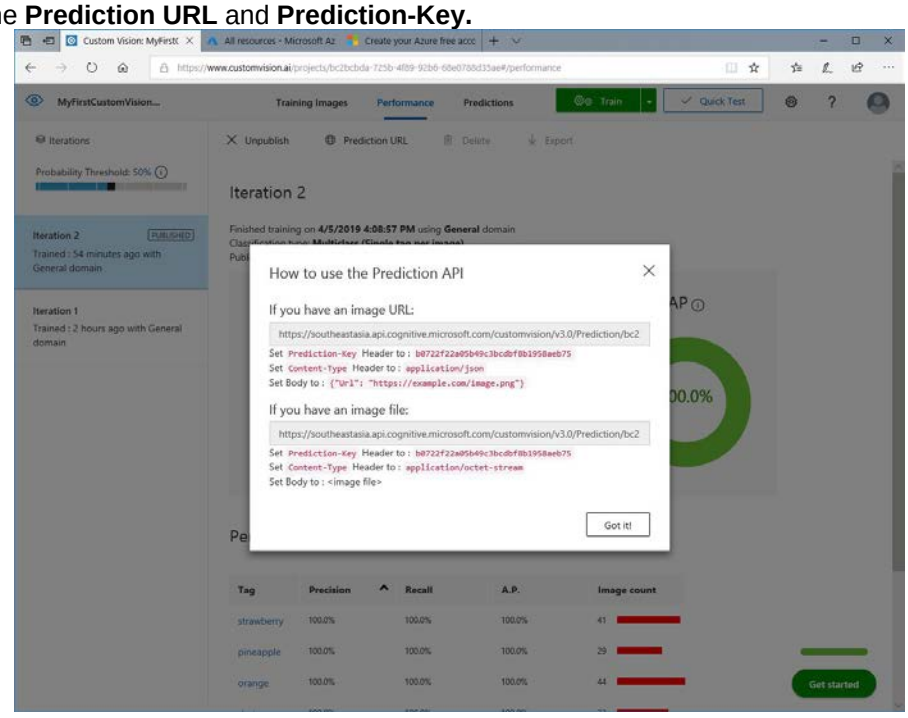


b) Once your model has been successfully published, you'll see a "Published" label appear next to your iteration

in the left-hand sidebar, and its name will appear in the description of the iteration.



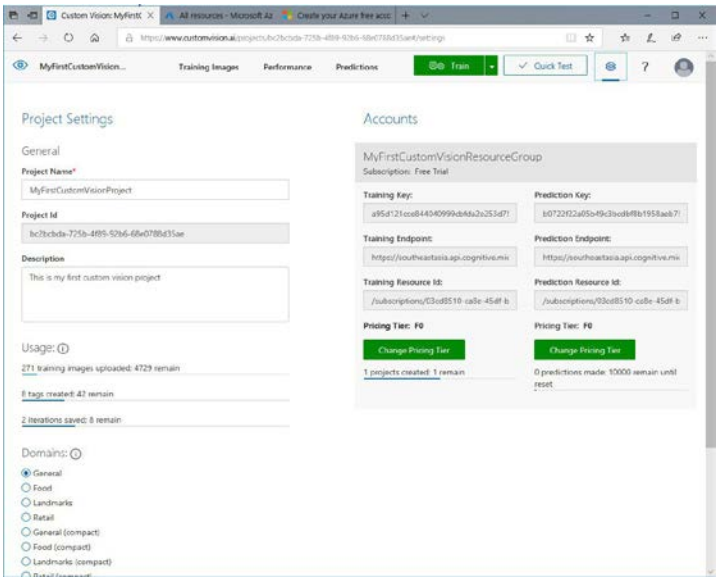
c) **Get the URL and prediction key**. Once your model has been published, you can retrieve the required information by selecting **Prediction URL**. This will open up a dialog with information for using the Prediction API, including the **Prediction URL** and **Prediction-Key.**



d) In this session, you will use a local image, so copy the URL under If you have an image file to a temporary location. Copy the corresponding Prediction-Key value as well.

e) **Install the custom Vision SDK** via “pip install azure-cognitiveservices-vision-customvision”

f) Get the training and prediction keys. The project needs a valid set of subscription keys in order to interact with the service. To get a set of free trial keys, go to the Custom Vision website and sign in with a Microsoft account. Select the gear icon in the upper right. In the Accounts section, see the values in the Training Key and Prediction Key fields. You will need these later.



g) **Create the prediction python script**. Add the following code to your script to create a new Custom Vision service project. Insert your subscription keys in the appropriate definitions.

```
01  from azure.cognitiveservices.vision.customvision.training import CustomVisionTrainingClient
02  from azure.cognitiveservices.vision.customvision.training.models import ImageFileCreateEntry
03  from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
04
04  ENDPOINT = "https://southeastasia.api.cognitive.microsoft.com"
05
06  # Replace with a valid key
07  training_key = "<your training key>"
08  prediction_key = "<your prediction key"
09  prediction_resource_id = “your prediction resource id"
10
11  publish_iteration_name = "<your model published iteration name"
12
13  # Now there is a trained endpoint that can be used to make a prediction
14  predictor = CustomVisionPredictionClient(prediction_key, endpoint=ENDPOINT)
15
16  with open("test_image.jpg", "rb") as image_contents:
17      results = predictor.classify_image("<your project id>", publish_iteration_name, image_contents.read())
18
19      # Display the results.
20      for prediction in results.predictions:
21          print ("\t" + prediction.tag_name + ": {0:.2f}%".format(prediction.probability * 100))
22
23
24
```

Run the above script to get the prediction from the model you trained.

```
(cv2_py35) C:\Users\kwseow\Projects\customvision>python customvision_predict.py
        coconut: 100.00%
        durian: 0.00%
        dragon fruit: 0.00%
        apple: 0.00%
        pineapple: 0.00%
        banana: 0.00%
        orange: 0.00%
        strawberry: 0.00%
```

Activity wrap-up:
We learn how to:
- ❑ Build a classifier through the Microsoft’s Custom Vision Service.
- ❑ Use a sample python script to use your classifier.

Activity 2 – Hands on creating a QnA chatbot

In this activity, we will learn:

- ❑ Create an Azure QnA Knowledge Base.
- ❑ Create an Azure Bot Service to use a knowledge base
- ❑ Chat with the bot to verify the code is working
- ❑ Link Telegram to bot's channel

1. Prerequisites

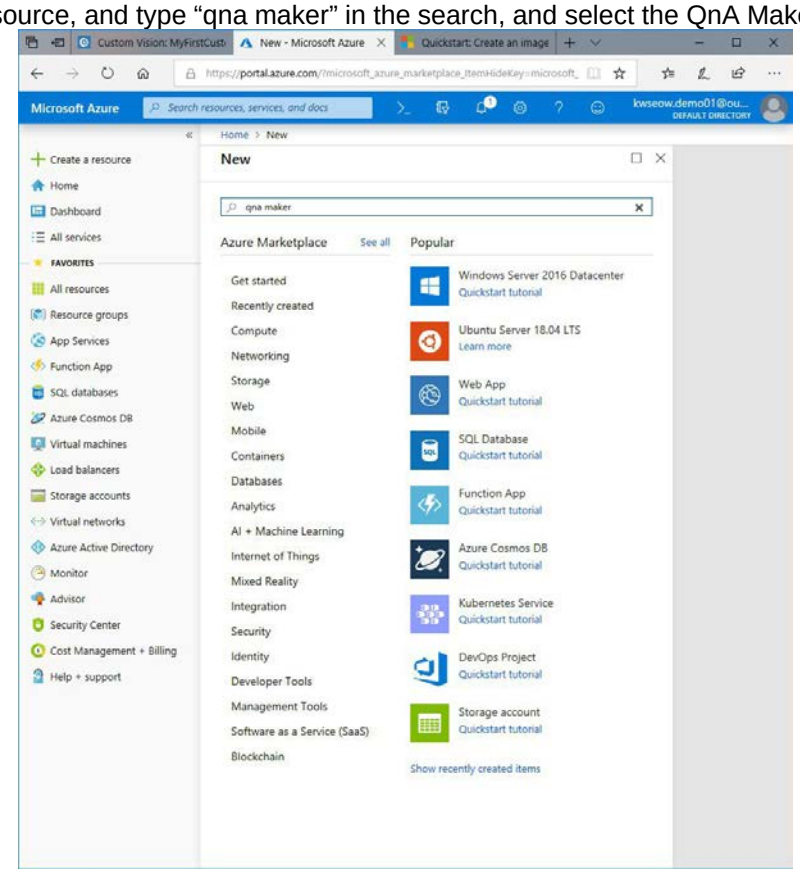
Before we can create a chatbot, we must first set up a QnA Maker Knowledge Base and a QnA Maker service in Azure.

a) QnA Maker Service

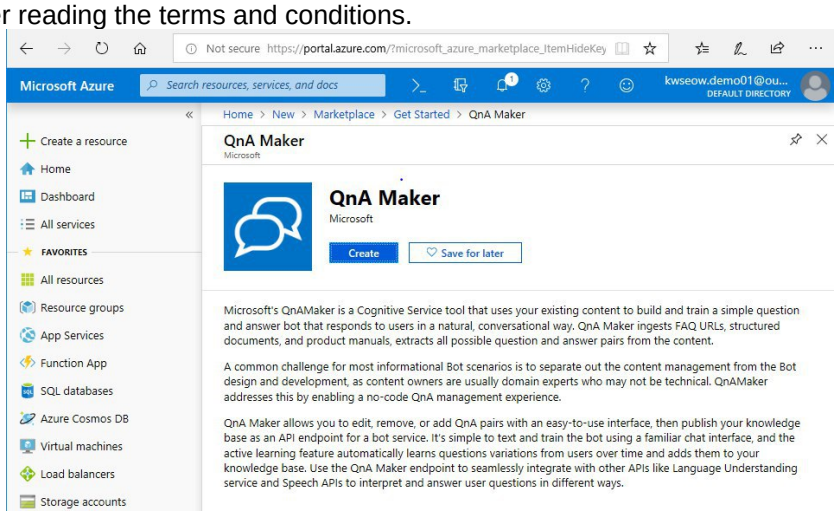
This procedure deploys a few Azure resources. Together, these resources manage the knowledge base content and provide question-answering capabilities though an endpoint.

i) Sign in to the Azure portal (https://portal.azure.com/)

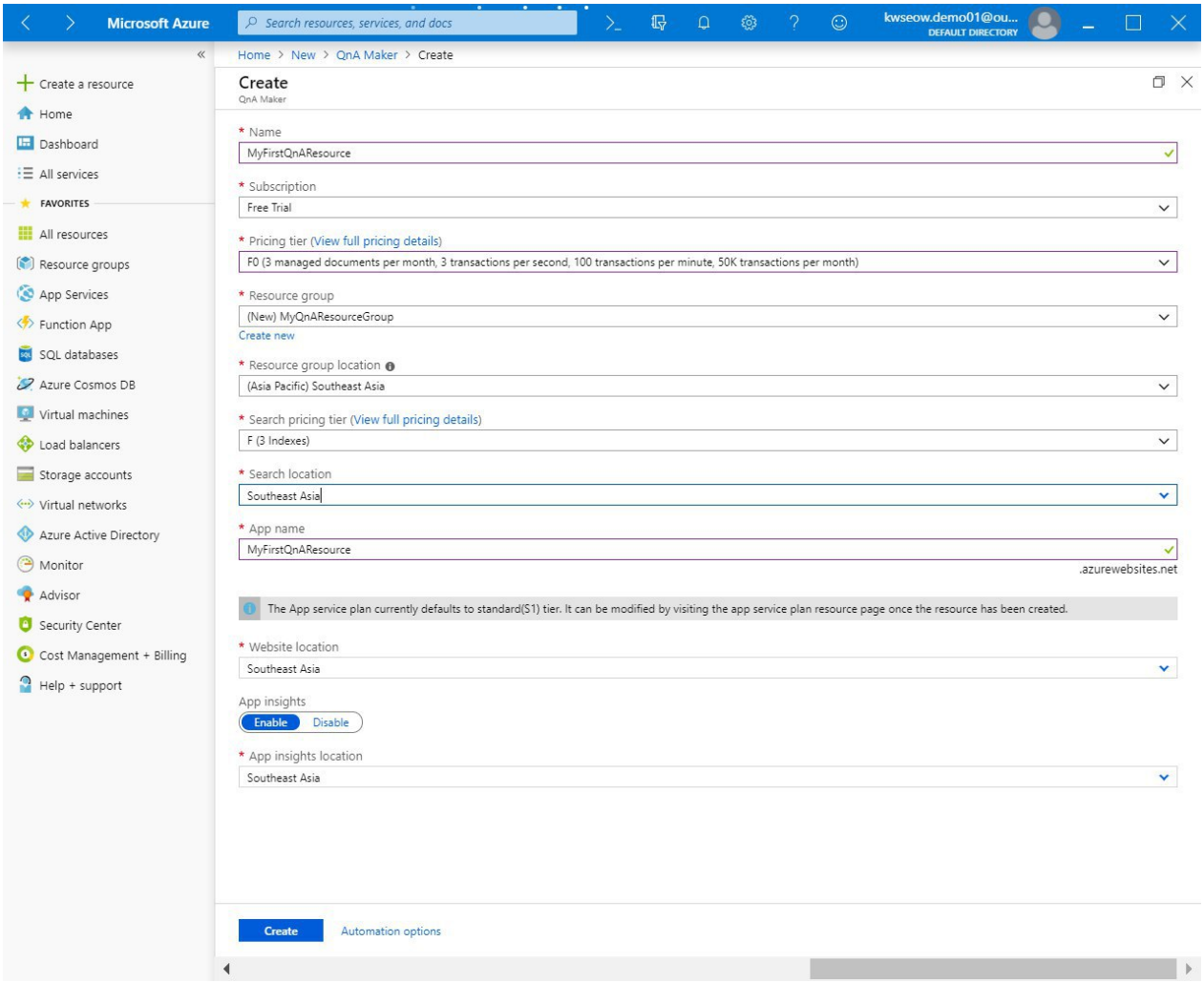
ii) Select Create a resource, and type "qna maker" in the search, and select the QnA Maker resource.



iii) Select Create after reading the terms and conditions.



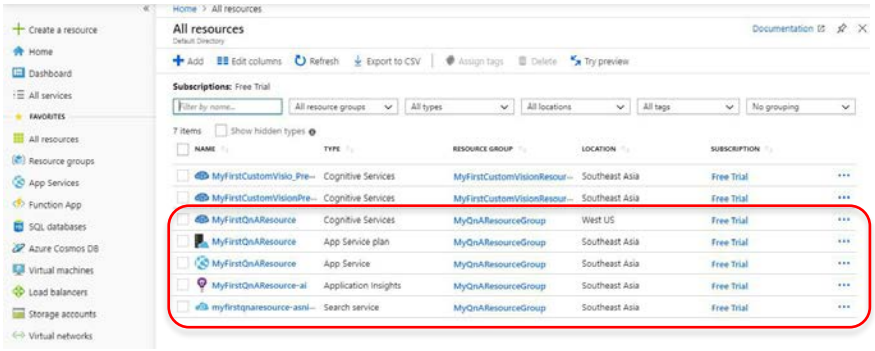
iv) In QnA Maker, select the appropriate tiers and regions.



- Fill the Name with a unique name to identify this QnA Maker service. This name also identifies the QnA Maker endpoint to which your knowledge bases will be associated.
- Choose the Subscription in which the QnA Maker resource will be deployed.
- Select the Management pricing tier for the QnA Maker management services (portal and management APIs). See here for details on the pricing of the SKUs.
- Create a new Resource Group (recommended) or use an existing one in which to deploy this QnA Maker resource. QnA Maker creates several Azure resources; when you create a resource group to hold these resources, you can easily find, manage, and delete these resources by the resource group name.
- Choose the Search pricing tier of the Azure Search service. If you see the Free tier option greyed out, it means you already have a Free Azure Search tier deployed in your subscription. In that case, you will need to start with the Basic Azure Search tier. See details of Azure search pricing here.
- Choose the Search Location where you want Azure Search data to be deployed. Restrictions in where customer data must be stored will inform the location you choose for Azure Search.
- Give a name to your App service in App name.
- By default the App service defaults to the standard (S1) tier. You can change the plan after creation. See more details of App service pricing here.
- Choose the Website location where the App Service will be deployed. Note: The Search Location can be different from the Website Location.
- Choose whether you want to enable Application Insights or not. If Application Insights is enabled, QnA Maker collects telemetry on traffic, chat logs, and errors.
- Choose the App insights location where Application Insights resource will be deployed.
- For cost savings measures, you can share some but not all Azure resources created for QnA Maker.

v) Once all the fields are validated, you can select **Create** to start deployment of these services in your subscription. It will take a few minutes to complete.

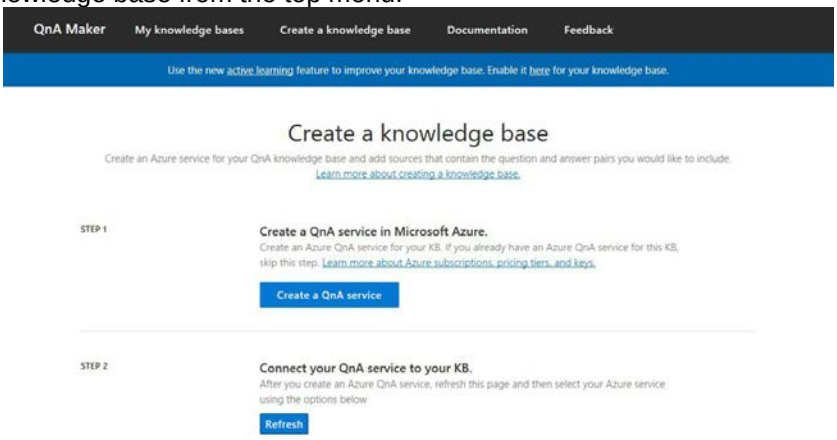
vi) Once the deployment is done, you will see the following resources created in your subscription.



2) QnA Maker Knowledge Base

a) Sign in to the QnA Maker portal (https://www.qnamaker.ai/).

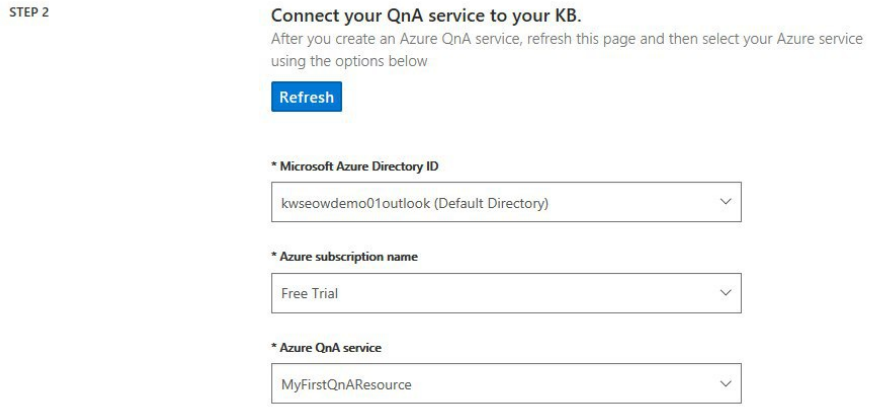
b) Select Create a knowledge base from the top menu.



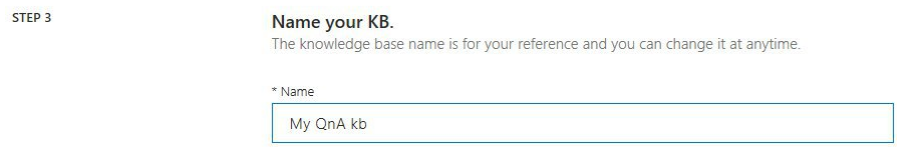
c) Skip the first step because you will use the QnA Maker service created in the previous step.

d) In the next step, select your existing setting:

Setting	Purpose
Microsoft Azure Directory Id	Your *Microsoft Azure Directory ID* is associated with the account you use to sign into the Azure portal and the QnA Maker portal.
Azure Subscription name	Your billing account you created the QnA Maker resource in.
Azure QnA Service	Your existing QnA Maker resource.

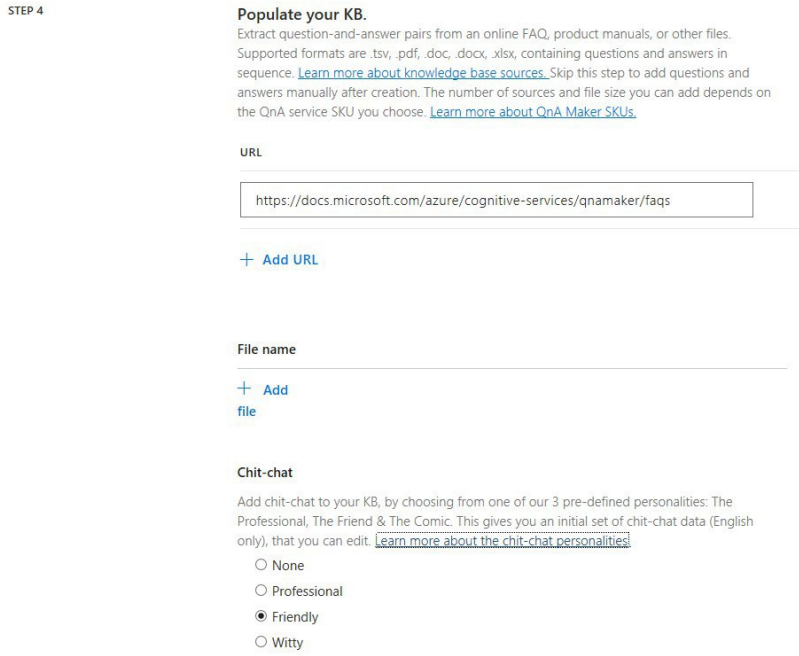


e) In the next step, Enter your knowledge base name, **My QnA kb**.



f) In the next step, populate your kb with the following settings. In our case, we will be using https://docs.microsoft.com/en-sg/azure/cognitive-services/qnamaker/faqs

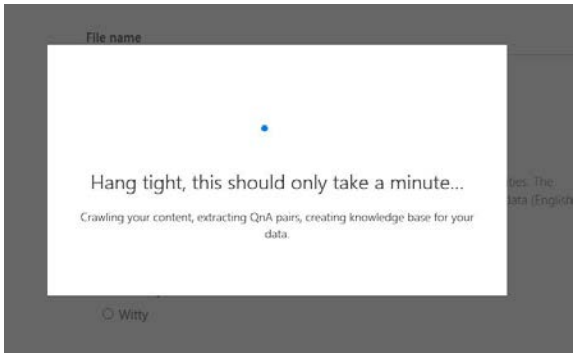
Setting name	Setting value	Purpose
Microsoft Azure Directory Id	Your *Microsoft Azure Directory ID* is associated with the account you use to sign into the Azure portal and the QnA Maker portal.	The contents of the FAQ at that URL are formatted with a question followed by an answer. QnA Maker can interpret this format to extract questions and the associated answers.
File	*not used in this activity*	This uploads files for questions and answers.
Chit-chat personality	Friendly	This gives a friendly and casual personality to common questions and answers. You can edit these questions and answers later.



Learn more about the chit-chat personalities (https://aka.ms/qnamaker-chitchat-kbs).

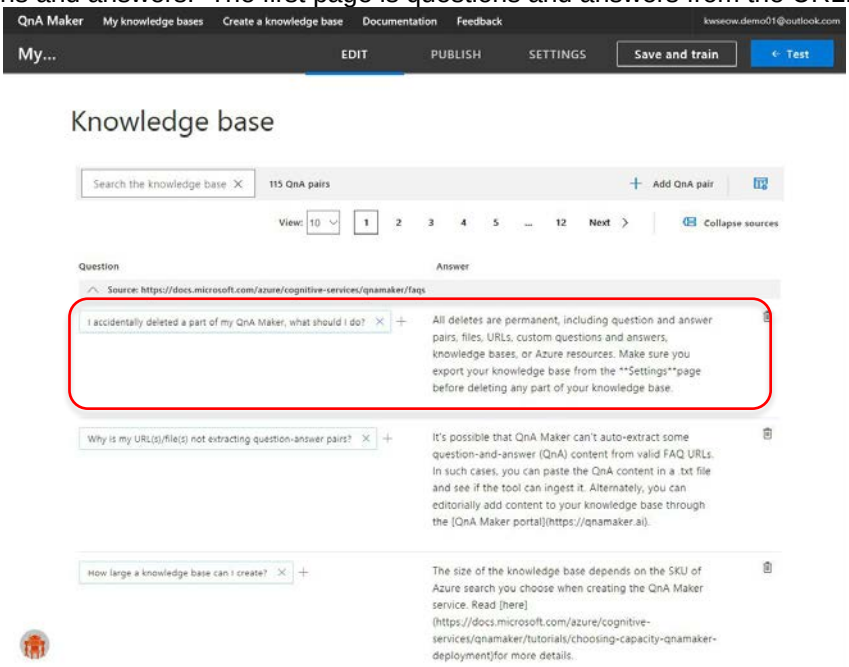
g) Select Create your KB to finish the creation process.

It will take a while to create the kb,

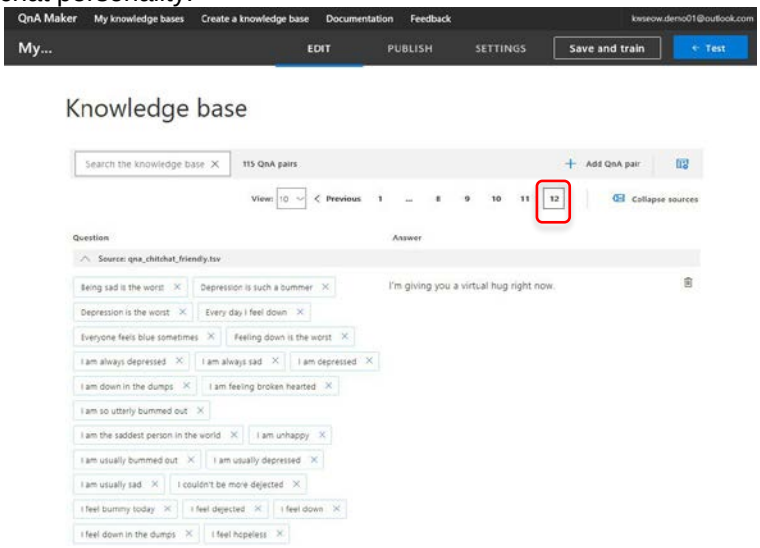


3) Review KB, save, and train

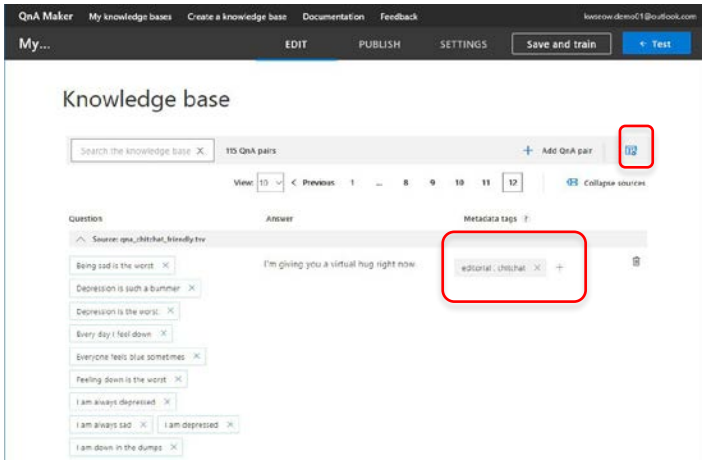
a) Review the questions and answers. The first page is questions and answers from the URL.



b) Select the last page of questions and answers from the bottom of the table. The page shows questions and answers from the Chit-chat personality.



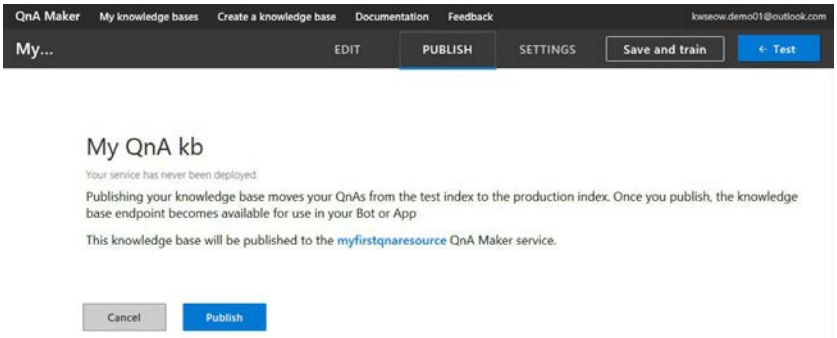
c) From the toolbar above the list of questions and answers, select the **metadata** icon. This shows the metadata tags for each question and answer. The Chit-chat questions have the **editorial: chit-chat** metadata already set. This metadata is returned to the client application along with the selected answer. The client application, such as a chat bot, can use this filtered metadata to determine additional processing or interactions with the user.



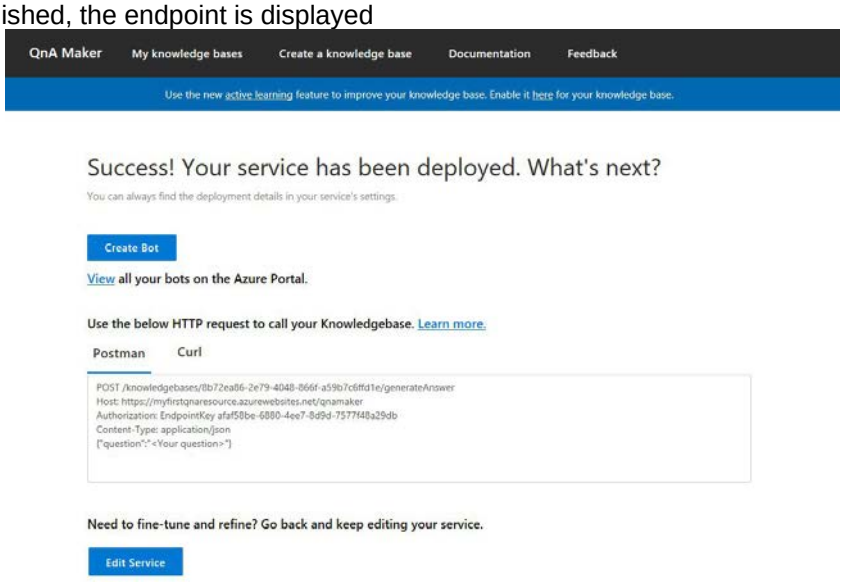
d) Select **Save and train** in the top menu bar.

4) Publish to get KB endpoints

a) Select the **Publish** button from the top menu. Once you are on the publish page, select **Publish**, next to the **Cancel** button.

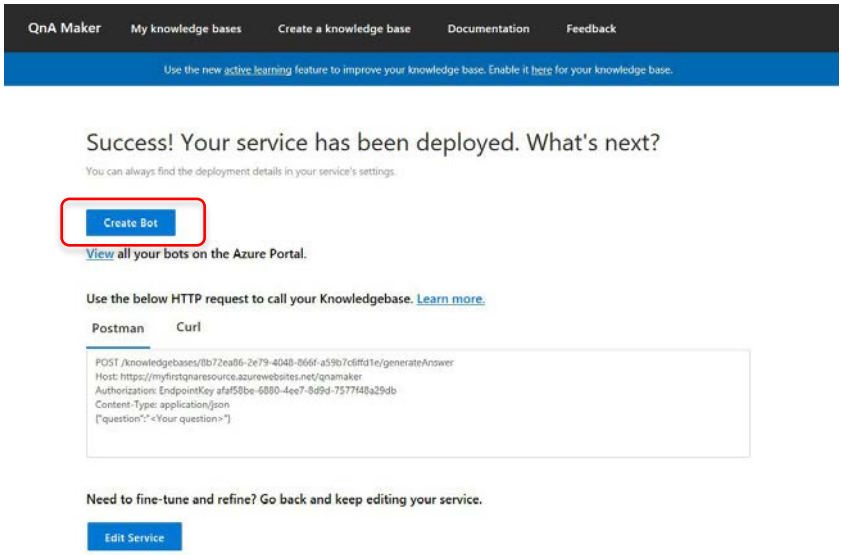


b) After the KB is published, the endpoint is displayed



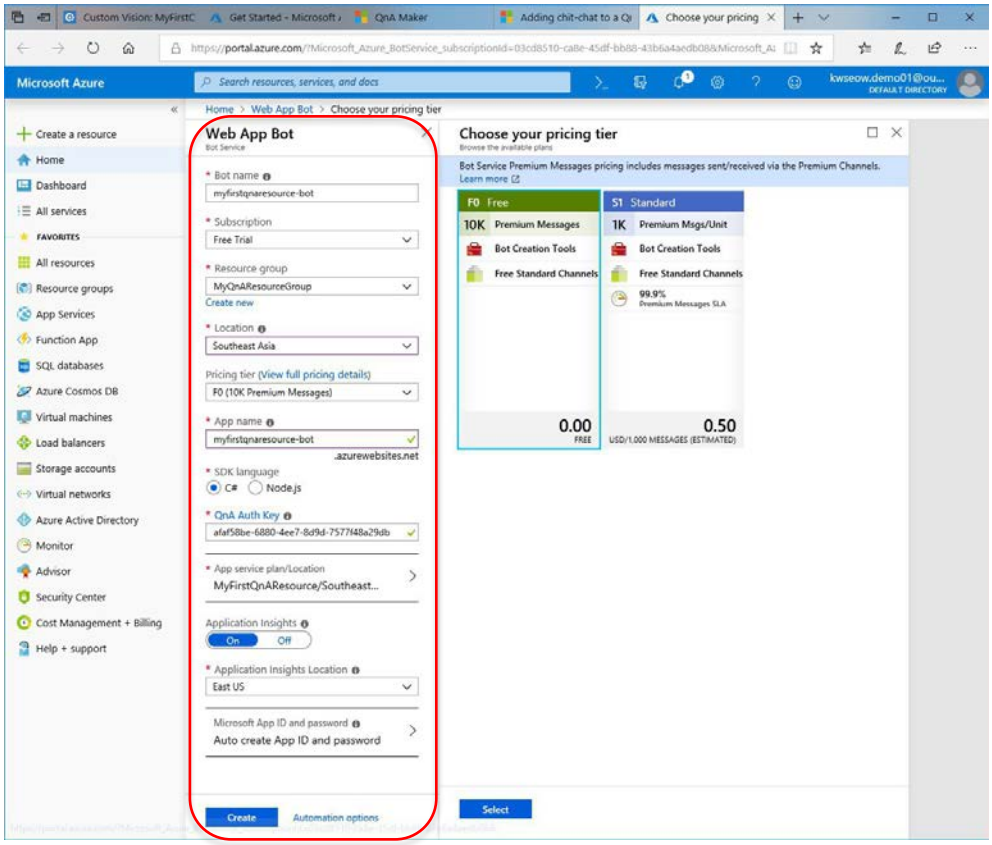
5) Create a QnA Bot

a) In the QnA Maker portal (go to the Publish page, and publish your knowledge base, if you have not done so). Select Create Bot.



b) Enter the settings to create the bot.

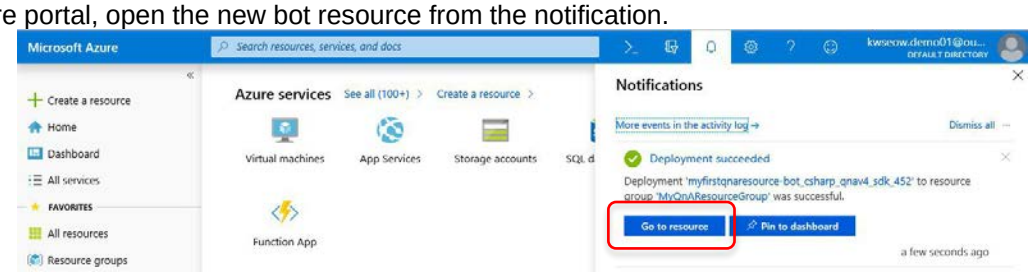
Setting	value	Purpose
Bot name	My-QnA-kb-bot.	This is the Azure resource name for the bot.
Subscription	See purpose	Select the same subscription as you used to create the QnA Maker resources.
Resource group	MyQnAResourceGroup	The resource group used for all the bot-related Azure resources.
Location	Southeast Asia	The bot's Azure resource location.
Pricing tier	F0	The free tier for the Azure bot service.
App name	Myfirstqnaresource-bot	This is a web app to support your bot only. This should not be the same app name as your QnA Maker service is already using. Sharing QnA Maker's web app with any other resource is not supported.
SDK Language	C#	This is the underlying programming language used by the bot framework SDK. Your choices are C# or Node.js.
QnA Auth Key	Do not change	This value is filled in for you.
App service plan/Location	Do not change	For this session, the location is not important.
Azure Storage	Do not change	Conversation data is stored in Azure Storage tables.
Application Insights	Do not change	Logging is sent to Application Insights.
Microsoft App ID	Do not change	Active directory user and password is required.



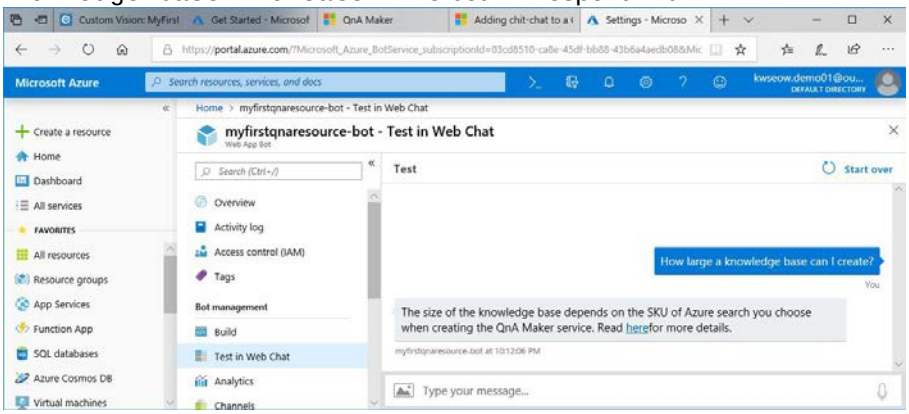
Wait a couple of minutes until the bot creation process notification reports success.

6) Chat with the Bot

a) In the Azure portal, open the new bot resource from the notification.



b) From **Bot management**, select **Test in Web Chat** and enter:
`How large a knowledge base I create?` The bot will respond with:



For more information about Azure Bots, see [Use QnA Maker to answer questions](https://docs.microsoft.com/en-sg/azure/bot-service/bot-builder-howto-qna?view=azure-bot-service-)
(https://docs.microsoft.com/en-sg/azure/bot-service/bot-builder-howto-qna?view=azure-bot-service-

4.0&tabs=cs).

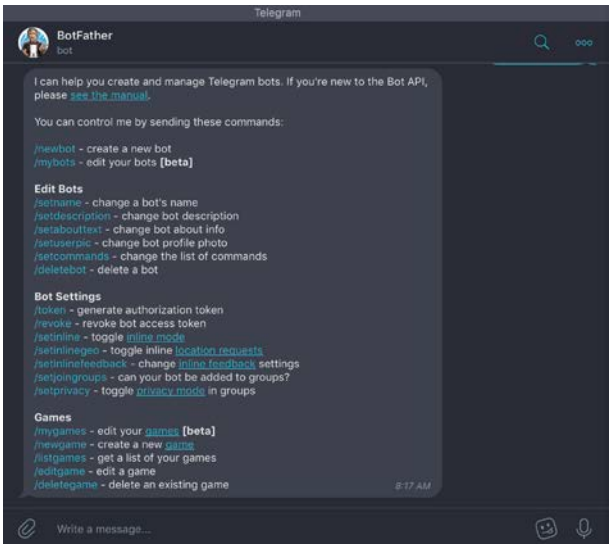
7) Connect your Bot to Telegram (channels)

A channel is a connection between the bot and communication apps. You configure a bot to connect to the channels you want it to be available on. The Bot Framework Service, configured through the Azure portal, connects your bot to these channels and facilitates communication between your bot and the user. You can connect to many popular services, such as Cortana, Facebook Messenger, Kik, and Slack, as well as several others. The Web Chat channel is pre-configured for you. In addition to standard channels provided with the Bot Connector Service, you can also connect your bot to your own client application using Direct Line as your channel.

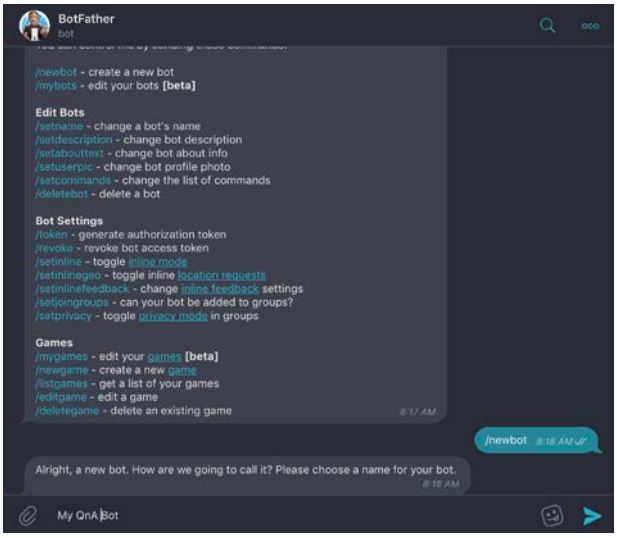
The Bot Framework Service allows you to develop your bot in a channel-agnostic way by normalizing messages that the bot sends to a channel. This involves converting it from the bot framework schema into the channel's schema. However, if the channel does not support all aspects of the bot framework schema, the service will try to convert the message to a format that the channel does support. For example, if the bot sends a message that contains a card with action buttons to the email channel, the connector may send the card as an image and include the actions as links in the message's text.

For most channels, you must provide channel configuration information to run your bot on the channel. Most channels require that your bot have an account on the channel, and others, like Facebook Messenger, require your bot to have an application registered with the channel also.

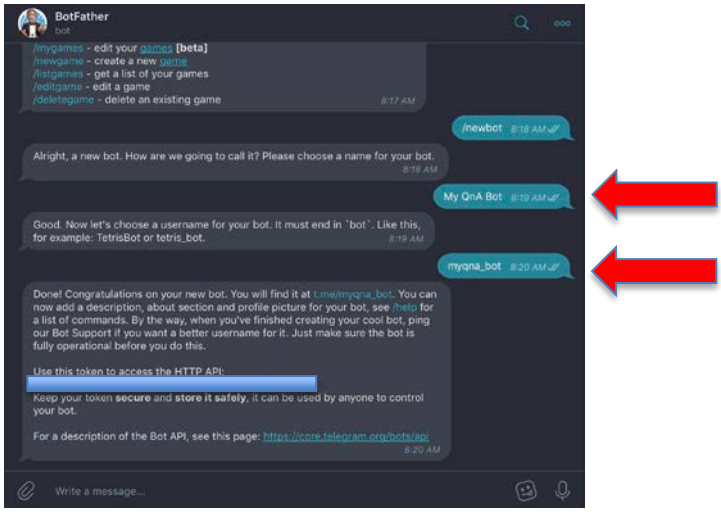
a) If you do not have telegram installed on your mobile phone, please install via Google Play store or Apple App Store.
b) If you have telegram installed on your laptop, use this link to connect to Bot Father (https://telegram.me/botfather) or add botfather. Otherwise, if you are using telegram on your mobile, start telegram and search for botfather.
c) Click on **Start** to start a conversation with botfather.



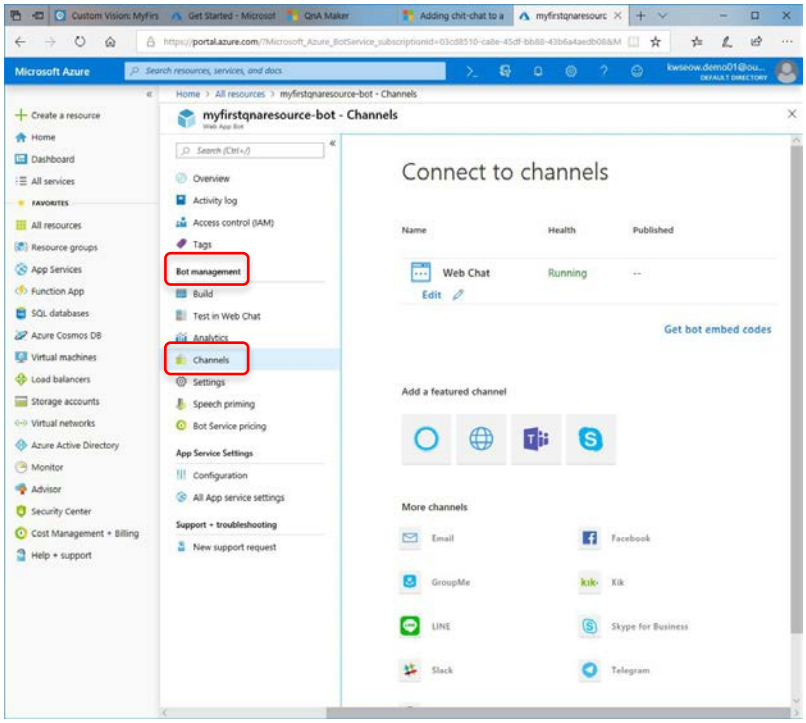
d) Create a new Telegram bot by sending command /newbot.



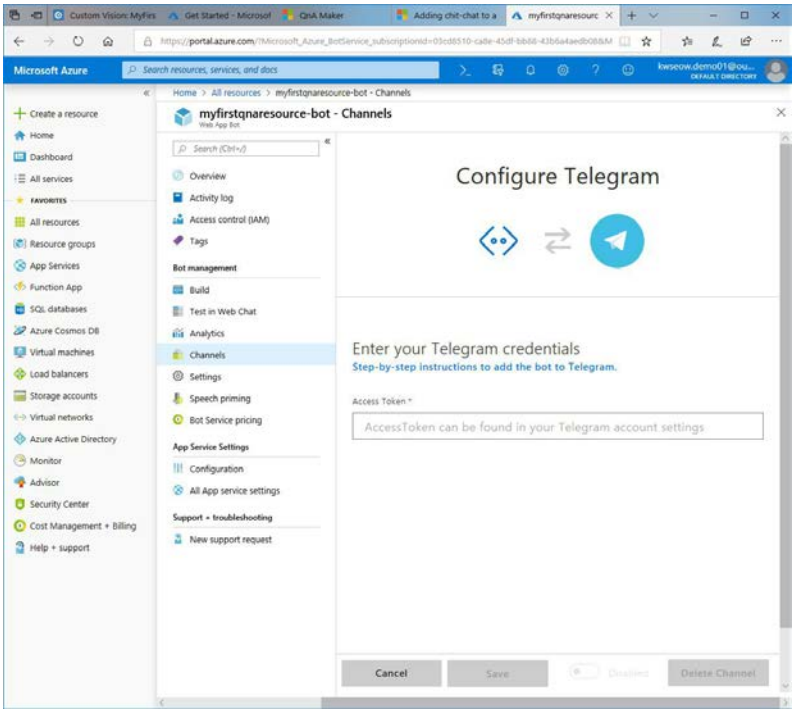
e) Give the bot a friendly name.
f) Specify a username.



g) Copy the Telegram bot's access token provide in the screen above. You will this info to configure the channel for the QnA bot
h) Sign in to the Azure Portal (https://portal.azure.com/).
i) Select the bot that you want to configure.
j) In the **Bot Service** blade, click Channels under **Bot Management**.

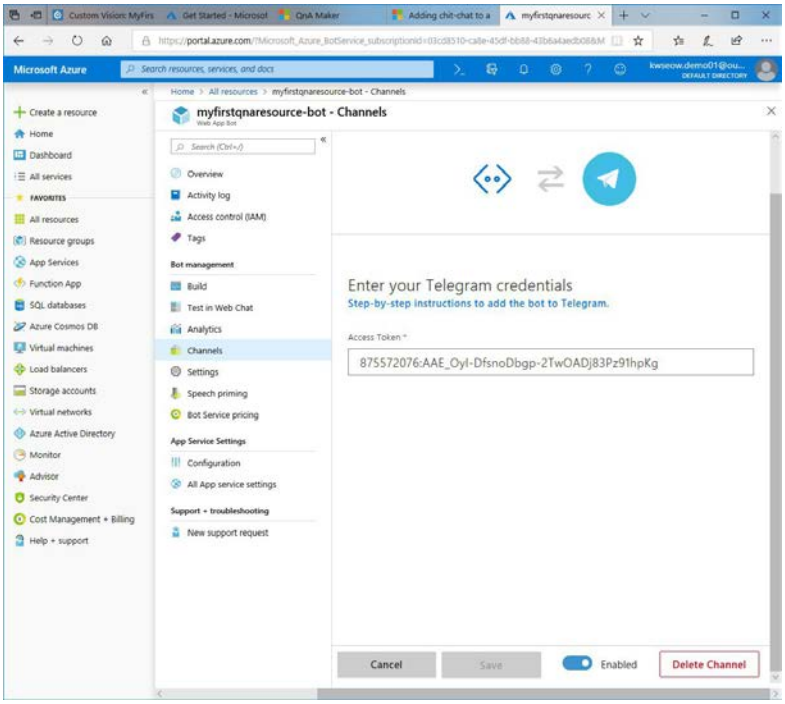


k) Click on the Telegram icon to add this channel.



l) Enter the code you copied in step 7.f and click **Save**.

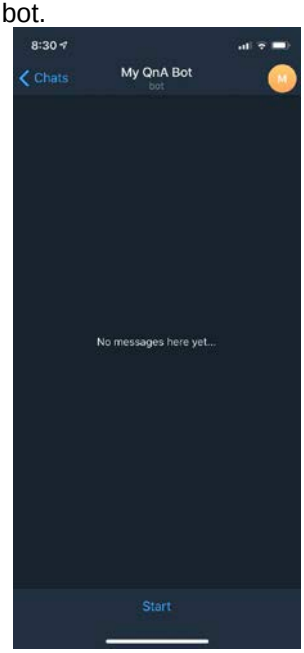
m) Make sure the Enable is switched on. When you have completed these steps, your bot will be successfully configured to communicate with users in Telegram.



8) Start chatting on Telegram

a) Launch telegram and add your bot. You can search for you by adding a @ to the username you used in step 7.e. For my case, *@myqna_bot*.

b) Click start to start a conversation with the bot.

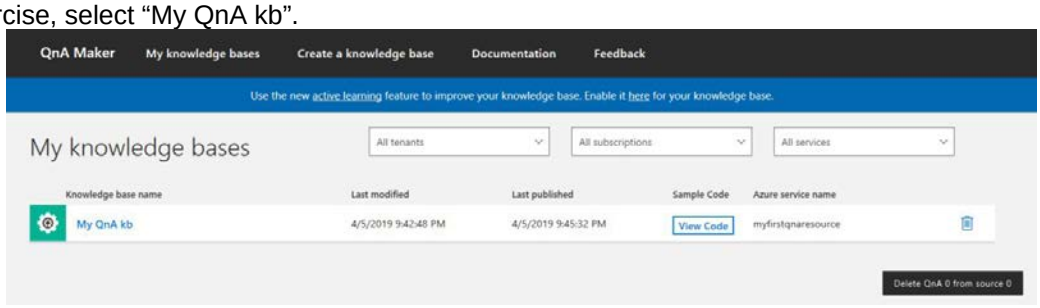


c) Start asking your bot!

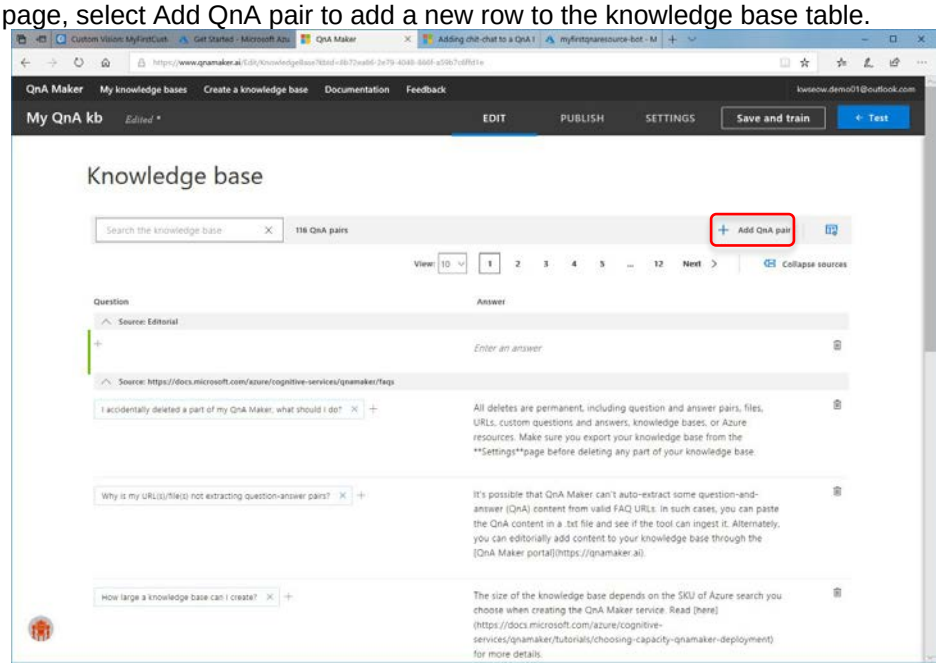
9) Edit a knowledge base (optional exercise)

a) Sign in to the QnA Maker portal (https://www.qnamaker.ai/).

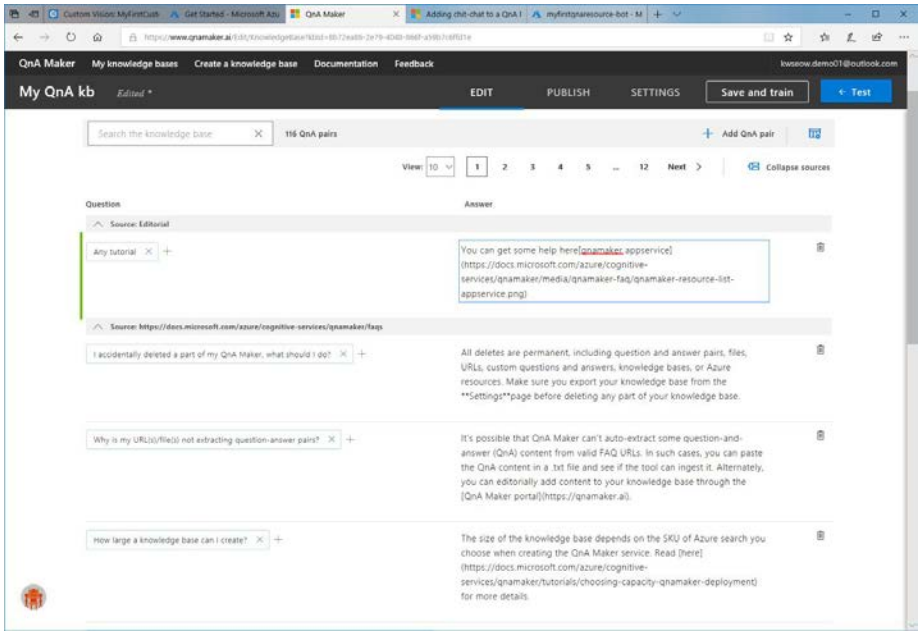
b) Select My Knowledge Bases in the top navigation bar. Select a particular knowledge base to make edits to it. In this exercise, select "My QnA kb".



c) On the Setting page, select Add QnA pair to add a new row to the knowledge base table.

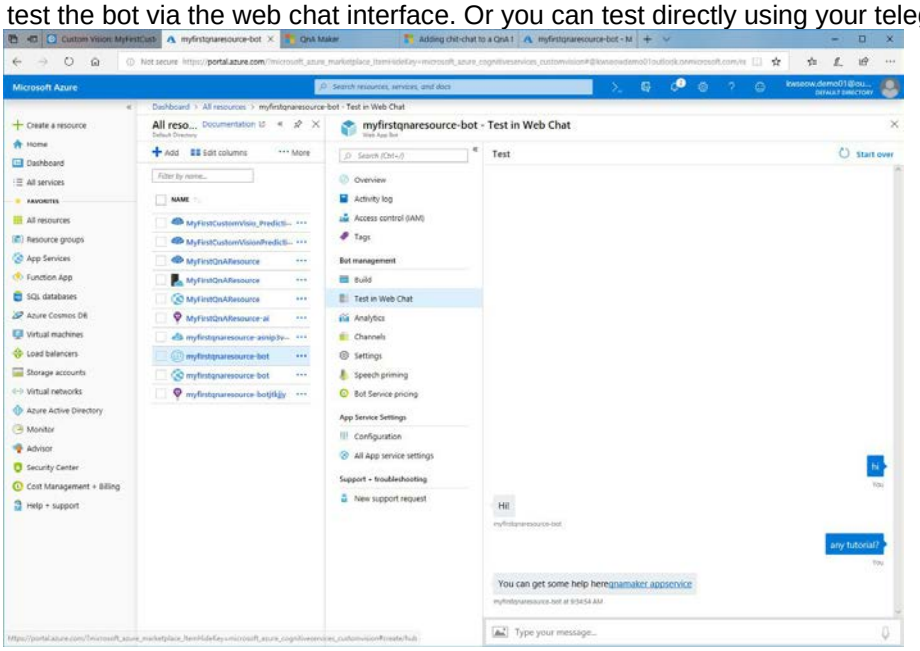


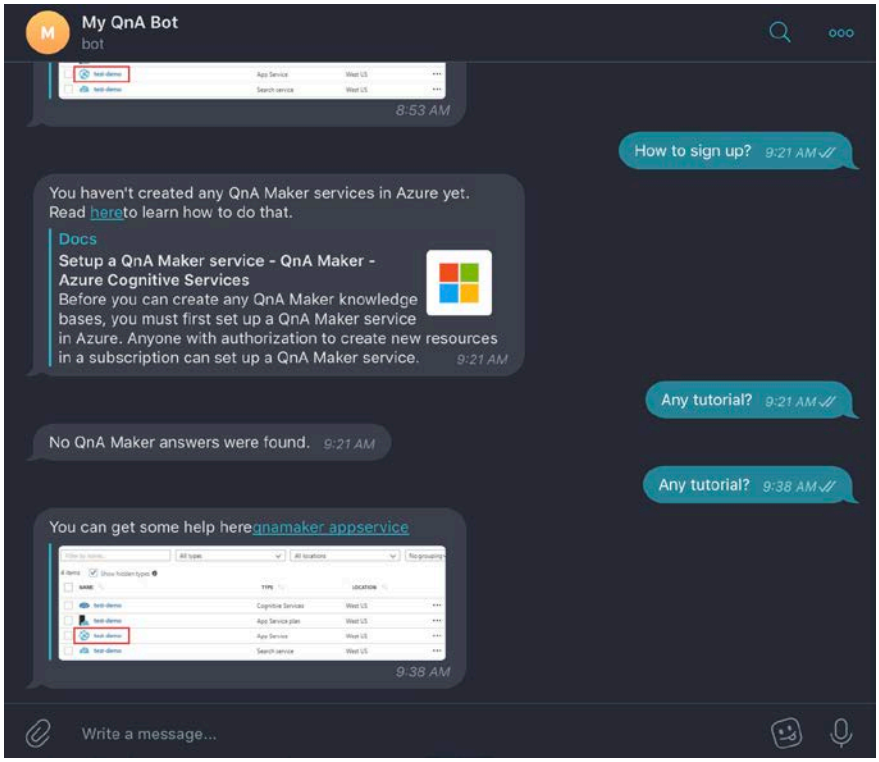
d) Add a QnA pair. When completed, click on Save and train to train the QnA Maker.



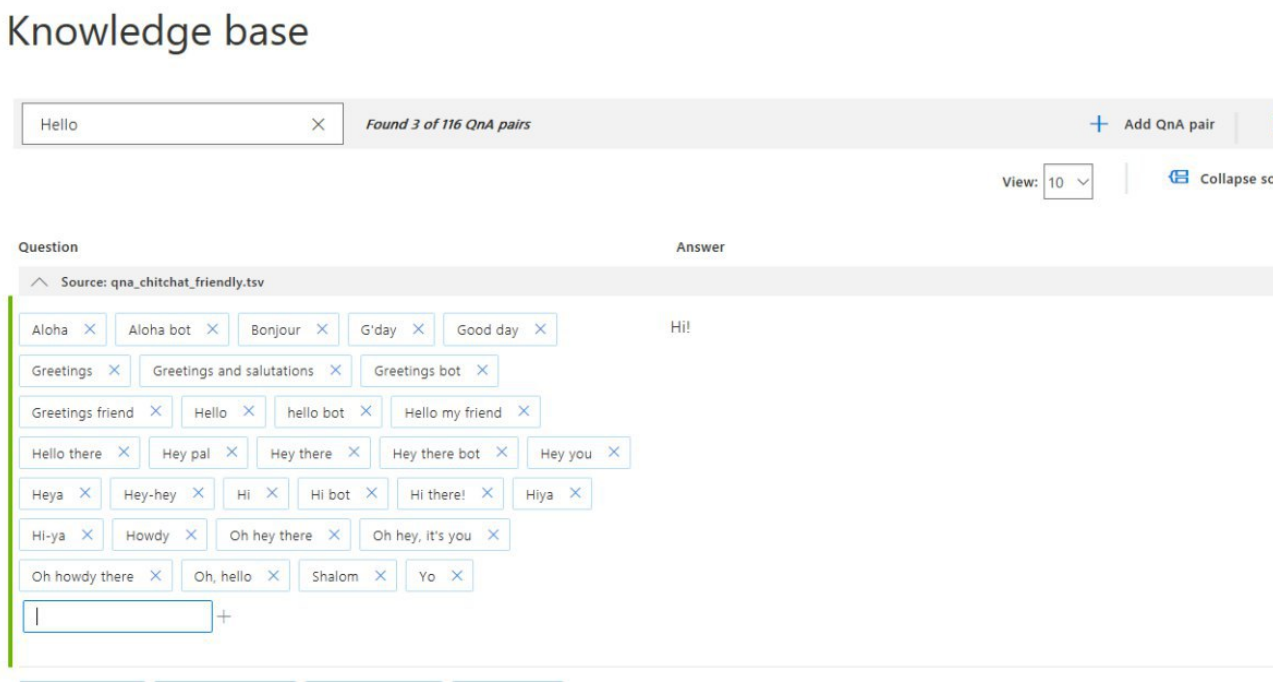
e) Follow step 4 to publish the knowledge base.

f) Repeat step 6 to test the bot via the web chat interface. Or you can test directly using your telegram bot too.





g) You can also add alternate questions to an existing QnA pair to improve the likelihood of a match to a user query.



Activity wrap-up:
We learn how to
- ❑ Create an Azure QnA Knowledge Base.
- ❑ Create an Azure Bot Service to use a knowledge base
- ❑ Chat with the bot to verify the code is working
- ❑ Link Telegram to bot's channel