

# Explore image classification

**Note** To complete this lab, you will need an [Azure subscription](#) in which you have administrative access.

The *Computer Vision* cognitive service provides useful pre-built models for working with images, but you'll often need to train your own model for computer vision. For example, suppose the Northwind Traders retail company wants to create an automated checkout system that identifies the grocery items customers want to buy based on an image taken by a camera at the checkout. To do this, you'll need to train a classification model that can classify the images to identify the item being purchased.

In Azure, you can use the **Custom Vision** cognitive service to train an image classification model based on existing images. There are two elements to creating an image classification solution. First, you must train a model to recognize different classes using existing images. Then, when the model is trained you must publish it as a service that can be consumed by applications.

To test the capabilities of the Custom Vision service, we'll use a simple command-line application that runs in the Cloud Shell. The same principles and functionality apply in real-world solutions, such as web sites or phone apps.

## Create a *Cognitive Services* resource

You can use the Custom Vision service by creating either a **Custom Vision** resource or a **Cognitive Services** resource.

**Note** Not every resource is available in every region. Whether you create a Custom Vision or Cognitive Services resource, only resources created in [certain regions](#) can be used to access Custom Vision services. For simplicity, a region is pre-selected for you in the configuration instructions below.

Create a **Cognitive Services** resource in your Azure subscription.

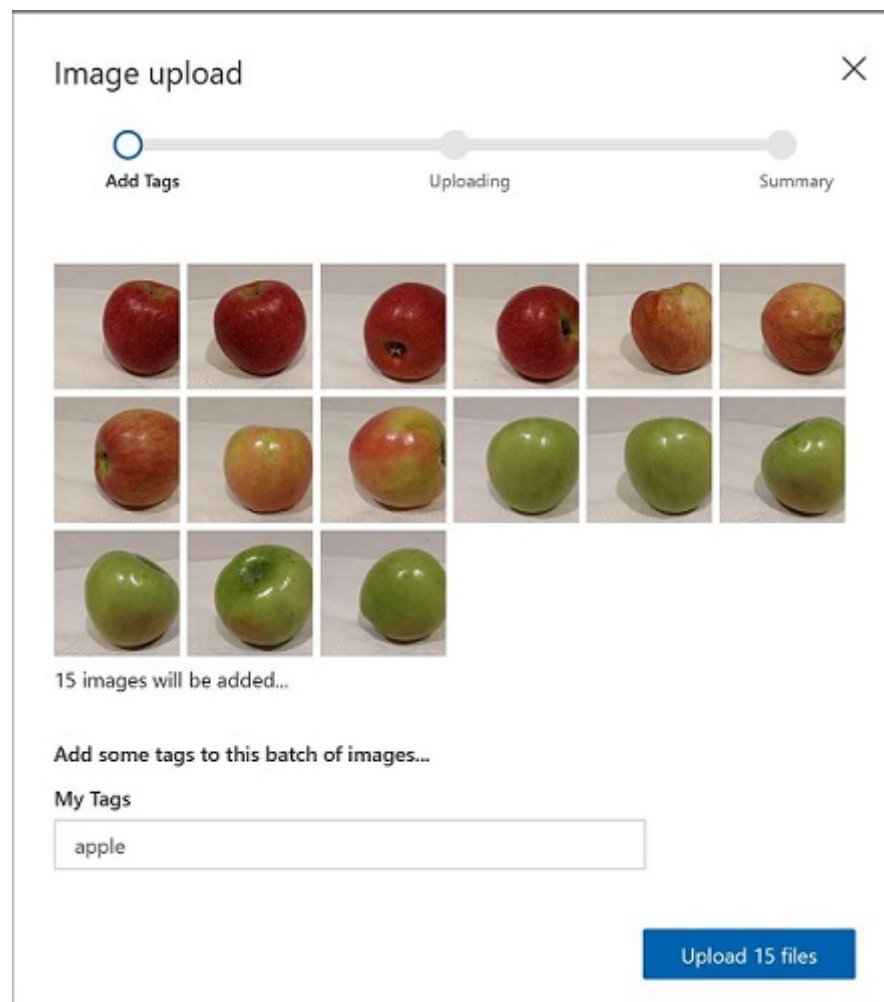
1. In another browser tab, open the Azure portal at <https://portal.azure.com>, signing in with your Microsoft account.
2. Click the **+ Create a resource** button, search for *Cognitive Services*, and create a **Cognitive Services** resource with the following settings:
  - **Subscription:** *Your Azure subscription.*
  - **Resource group:** *Select or create a resource group with a unique name.*
  - **Region:** East US
  - **Name:** *Enter a unique name.*
  - **Pricing tier:** Standard S0
  - **By checking this box I acknowledge that I have read and understood all the terms below:** Selected.
3. Review and create the resource, and wait for deployment to complete. Then go to the deployed resource.
4. View the **Keys and Endpoint** page for your Cognitive Services resource. You will need the endpoint and keys to connect from client applications.

## Create a Custom Vision project

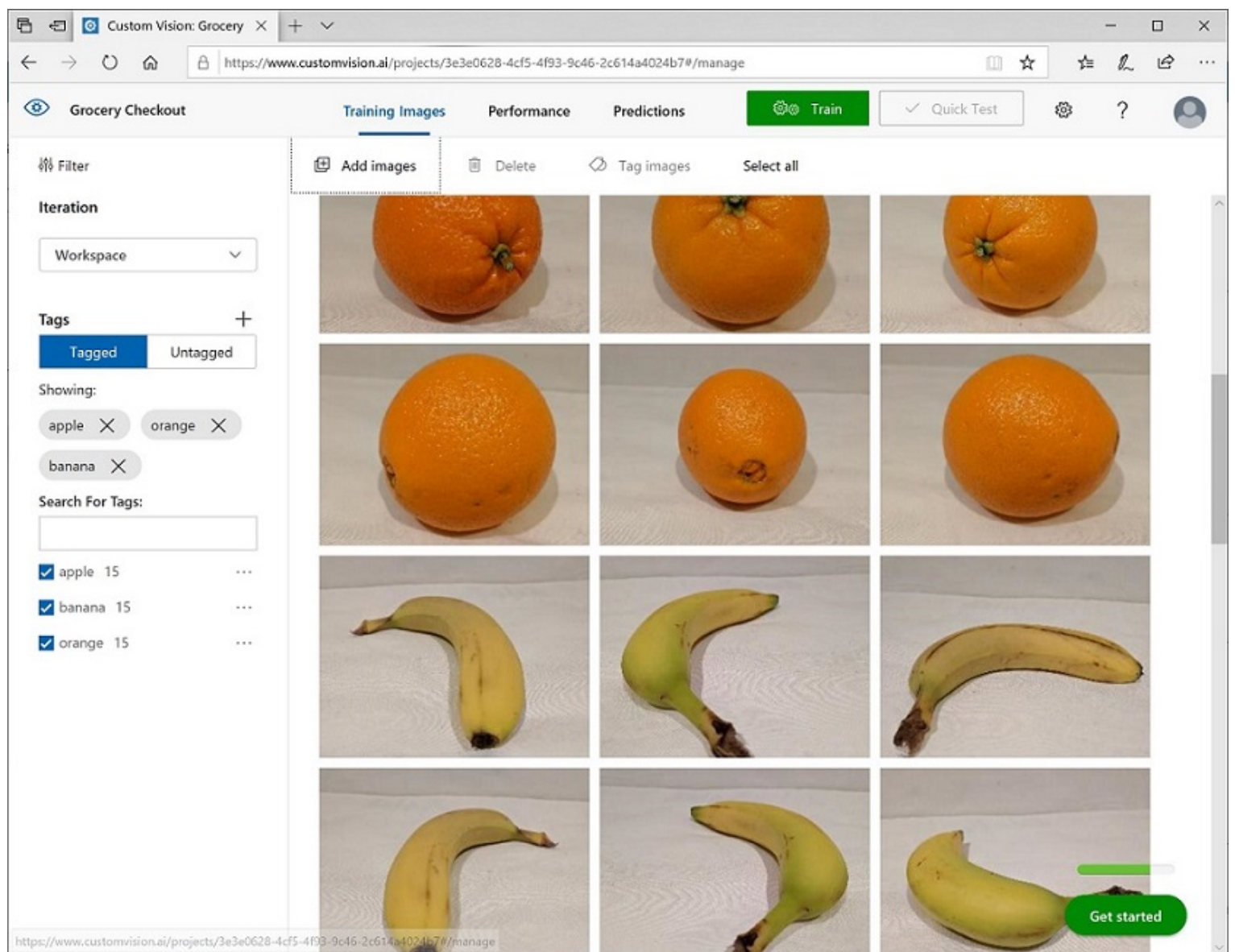
To train an object detection model, you need to create a Custom Vision project based on your training resource. To do this, you'll use the Custom Vision portal.

1. Download and extract the training images from <https://aka.ms/fruit-images>. These images are provided in a zipped folder, which when extracted contains subfolders called **apple**, **banana**, and **orange**.

2. In another browser tab, open the Custom Vision portal at <https://customvision.ai>. If prompted, sign in using the Microsoft account associated with your Azure subscription and agree to the terms of service.
3. In the Custom Vision portal, create a new project with the following settings:
  - **Name:** Grocery Checkout
  - **Description:** Image classification for groceries
  - **Resource:** *The Custom Vision resource you created previously*
  - **Project Types:** Classification
  - **Classification Types:** Multiclass (Single tag per image)
  - **Domains:** Food
4. Click **Add images**, and select all of the files in the **apple** folder you extracted previously. Then upload the image files, specifying the tag *apple*, like this:



5. Repeat the previous step to upload the images in the **banana** folder with the tag *banana*, and the images in the **orange** folder with the tag *orange*.
6. Explore the images you have uploaded in the Custom Vision project - there should be 15 images of each class, like this:

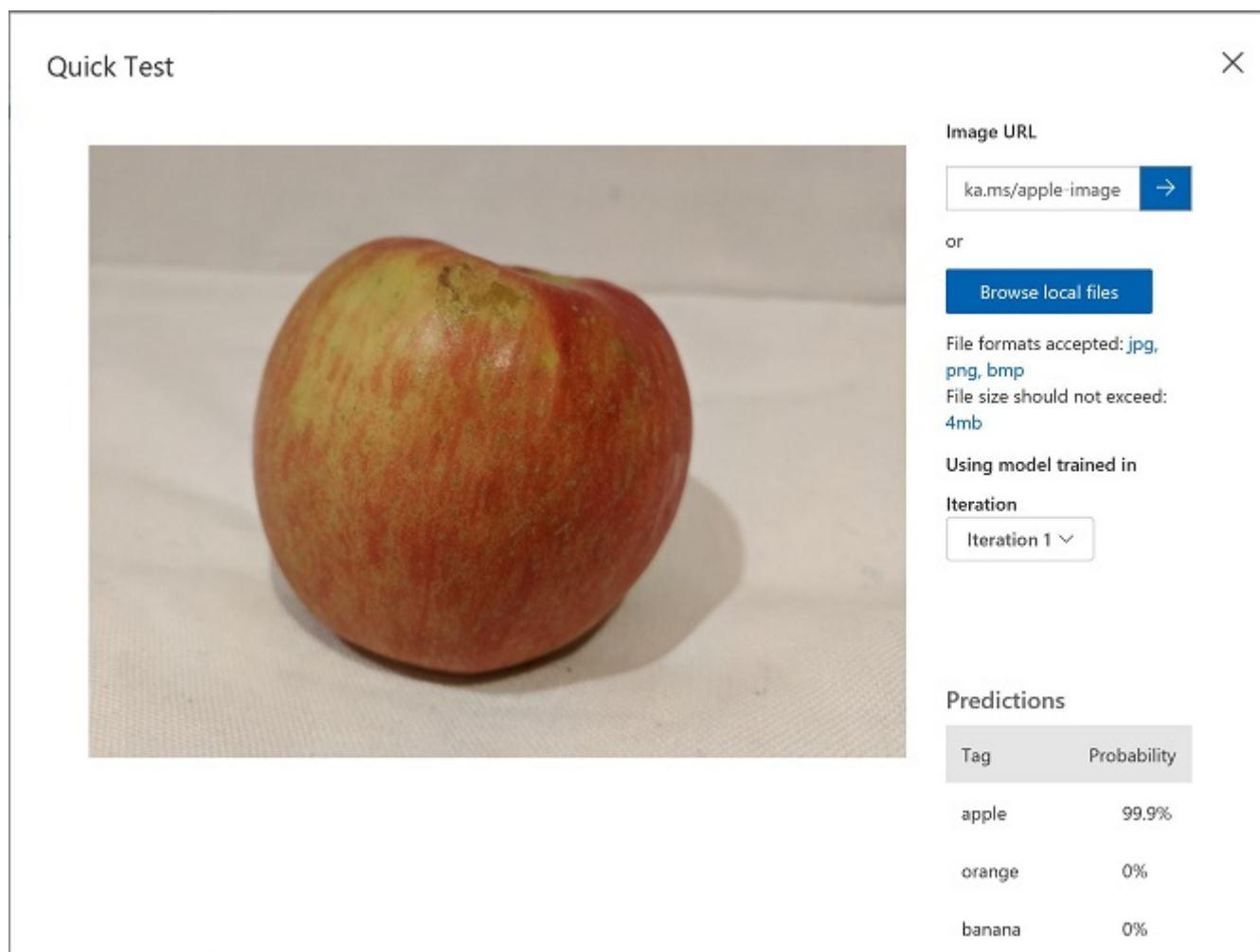


7. In the Custom Vision project, above the images, click **Train** to train a classification model using the tagged images. Select the **Quick Training** option, and then wait for the training iteration to complete (this may take a minute or so).
8. When the model iteration has been trained, review the *Precision*, *Recall*, and *AP* performance metrics - these measure the prediction accuracy of the classification model, and should all be high.

## Test the model

Before publishing this iteration of the model for applications to use, you should test it.

1. Above the performance metrics, click **Quick Test**.
2. In the **Image URL** box, type `https://aka.ms/apple-image` and click →
3. View the predictions returned by your model - the probability score for *apple* should be the highest, like this:



4. Close the **Quick Test** window.

## Publish the image classification model

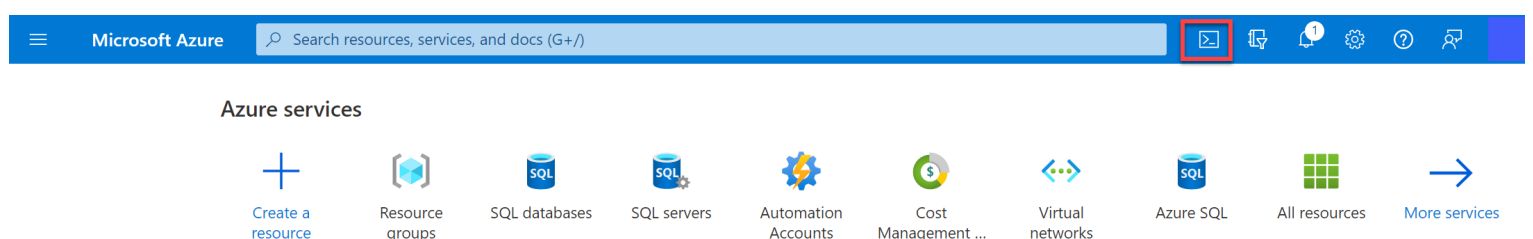
Now you're ready to publish your trained model and use it from a client application.

1. Click **Publish** to publish the trained model with the following settings:
  - **Model name:** groceries
  - **Prediction Resource:** *The prediction resource you created previously.*
2. After publishing, click the *Prediction URL* (🌐) icon to see information required to use the published model. Later, you will need the appropriate URL and Prediction-Key values to get a prediction from an Image URL, so keep this dialog box open and carry on to the next task.

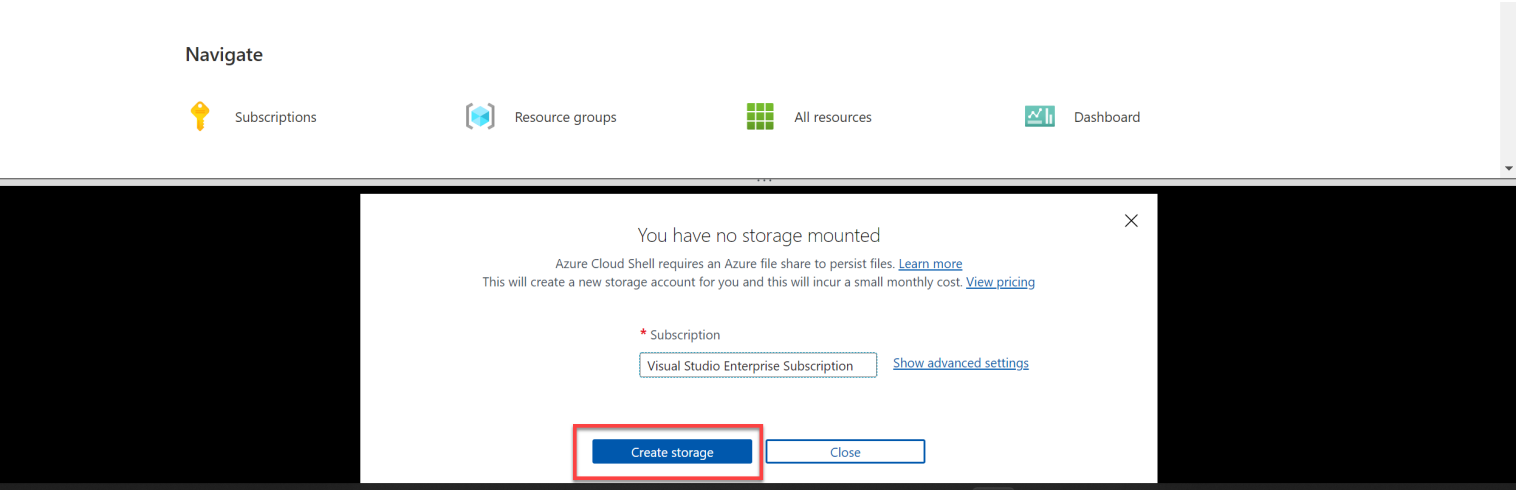
## Run Cloud Shell

To test the capabilities of the Custom Vision service, we'll use a simple command-line application that runs in the Cloud Shell on Azure.

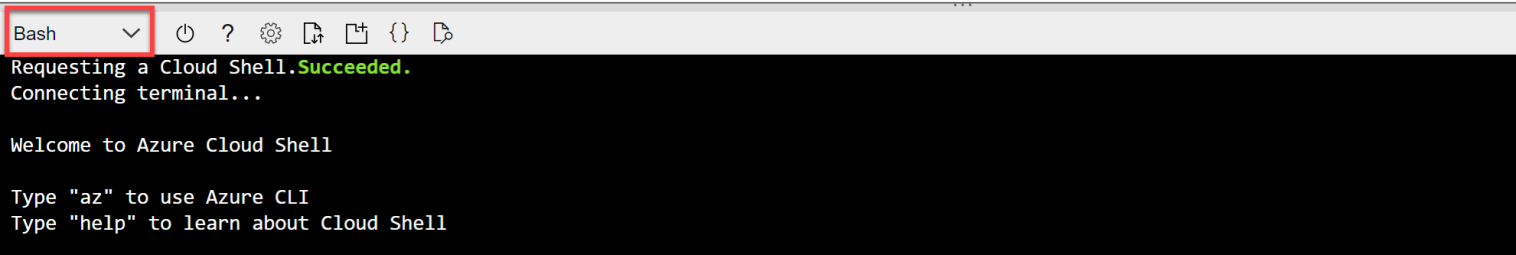
1. In the Azure portal, select the **[>\_]** (*Cloud Shell*) button at the top of the page to the right of the search box. This opens a Cloud Shell pane at the bottom of the portal.



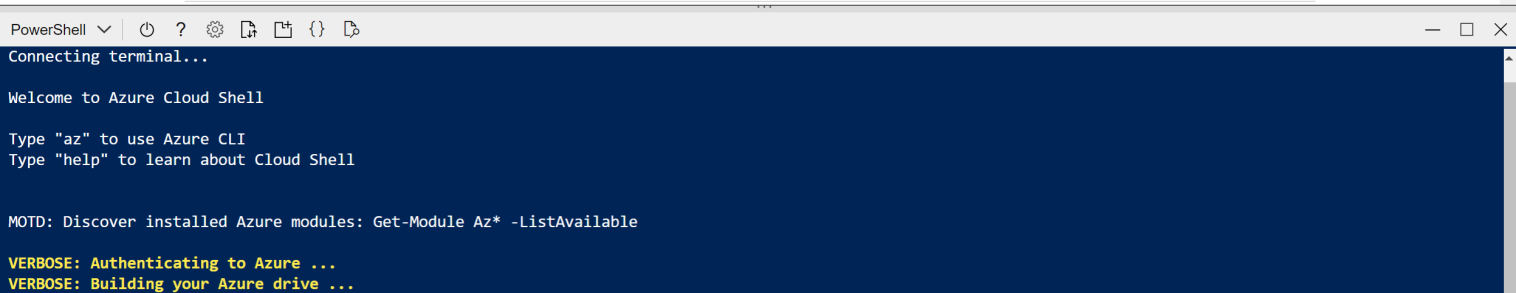
2. The first time you open the Cloud Shell, you may be prompted to choose the type of shell you want to use (*Bash* or *PowerShell*). Select **PowerShell**. If you do not see this option, skip the step.
3. If you are prompted to create storage for your Cloud Shell, ensure your subscription is specified and select **Create storage**. Then wait a minute or so for the storage to be created.



4. Make sure the the type of shell indicated on the top left of the Cloud Shell pane is switched to *PowerShell*. If it is *Bash*, switch to *PowerShell* by using the drop-down menu.



5. Wait for PowerShell to start. You should see the following screen in the Azure portal:



## Configure and run a client application

Now that you have a Cloud Shell environment, you can run a simple application that uses the Custom Vision service to analyze an image.

1. In the command shell, enter the following command to download the sample application and save it to a folder called ai-900.

Code Copy

```
git clone https://github.com/MicrosoftLearning/AI-900-AIFundamentals ai-900
```

**Tip** If you already used this command in another lab to clone the *ai-900* repository, you can skip this step.

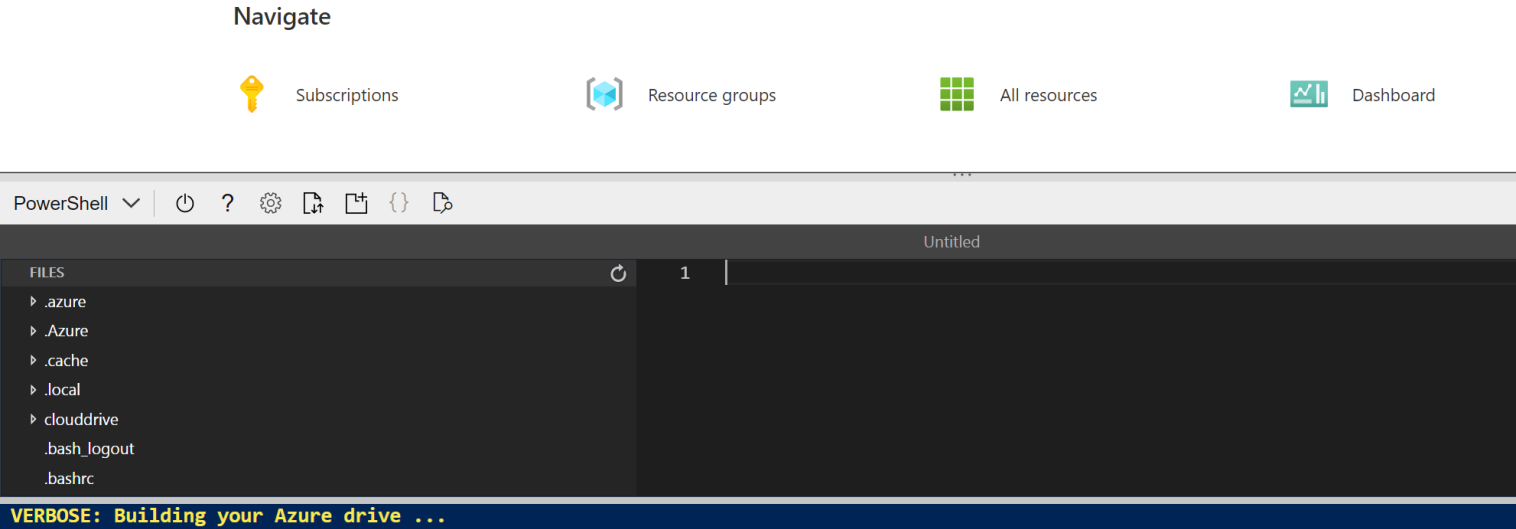
2. The files are downloaded to a folder named **ai-900**. Now we want to see all of the files in your Cloud Shell storage and work with them. Type the following command into the shell:

Code Copy

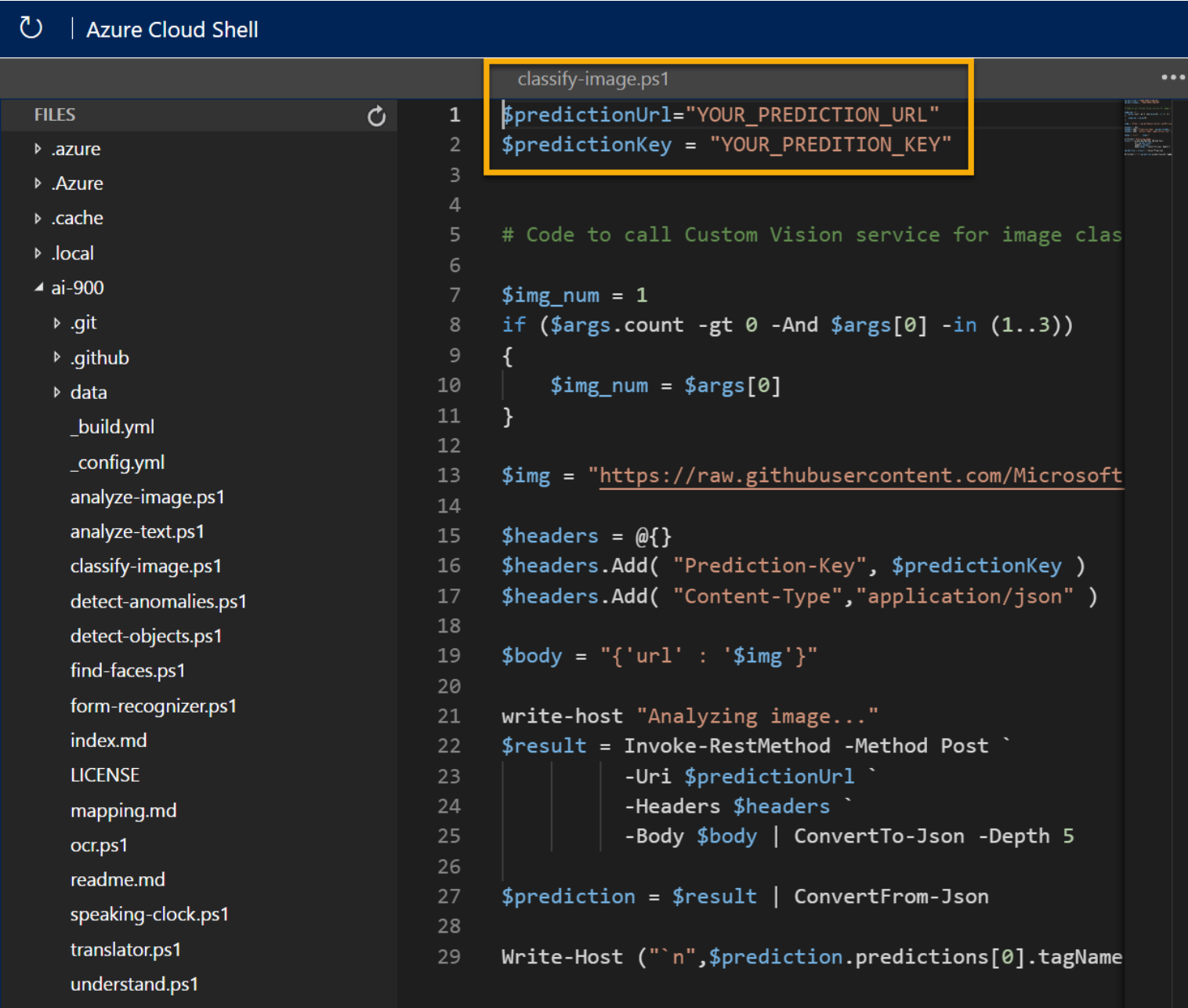
```
code .
```

Notice how this opens up an editor like the one in the image below:






3. In the **Files** pane on the left, expand **ai-900** and select **classify-image.ps1**. This file contains some code that uses the Custom Vision model to analyze an image, as shown here:

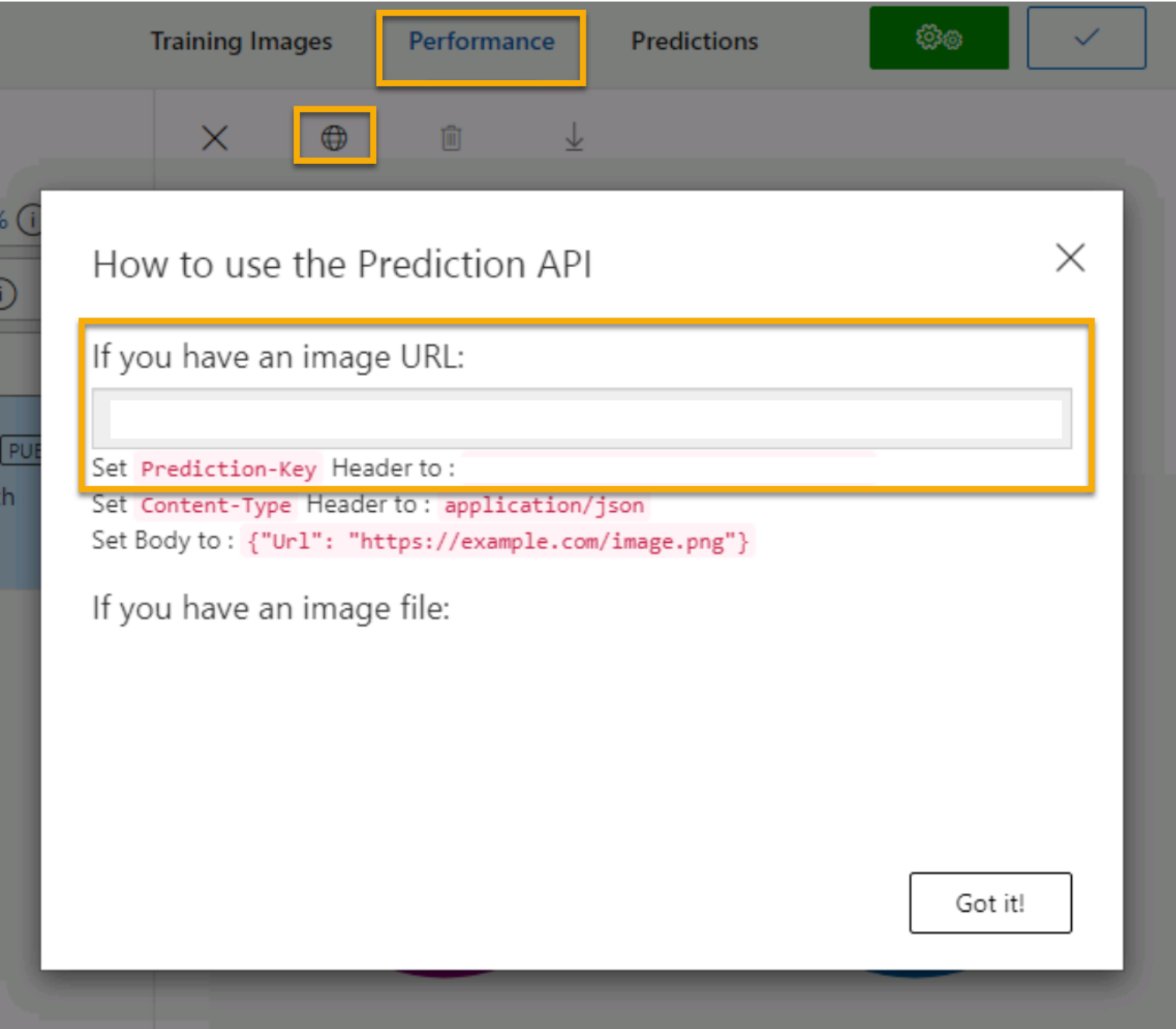


4. Don't worry too much about the details of the code, the important thing is that it needs the prediction URL and key for your Custom Vision model when using an image URL.

Get the *prediction URL* from the dialog box in your Custom Vision project.

 **Note** Remember, you reviewed the *prediction URL* after you published the image classification model. To find the *prediction URL*, navigate to the **Performance** tab in your project, then click on **Prediction URL** (if the screen is compressed, you may just see a globe icon). A dialog box will appear. Copy the url for **If you have an image URL**. Paste it into the code editor, replacing **YOUR\_PREDICTION\_URL**.

Using the same dialog box, get the *prediction key*. Copy the prediction key displayed after *Set Prediction-Key Header to*. Paste it in the code editor, replacing the **YOUR\_PREDICTION\_KEY** placeholder value.



After pasting the Prediction URL and Prediction Key values, the first two lines of code should look similar to this:

Code Copy

```
$predictionUrl="https..."  
$predictionKey ="1a2b3c4d5e6f7g8h9i0j...."
```

- 5. At the top right of the editor pane, use the ... button to open the menu and select **Save** to save your changes. Then open the menu again and select **Close Editor**.
- You will use the sample client application to classify several images into the apple, banana, or orange category.
- 6. We will classify this image:

[Test the model](#)

[Publish the image classification model](#)

[Run Cloud Shell](#)

Configure and run a client application

[Learn more](#)



In the PowerShell pane, enter the following commands to run the code:

Code	Copy
<pre>cd ai-900 ./classify-image.ps1 1</pre>	

- 7. Review the prediction, which should be **apple**.
- 8. Now let’s try another image:



Run this command:

Code	Copy
------	------



```
./classify-image.ps1 2
```

9. Verify that the model classifies this image as **banana**.

10. Finally, let's try the third test image:



Run this command:

```
Code
```

 Copy

```
./classify-image.ps1 3
```

11. Verify that the model classifies this image as **orange**.

## Learn more

This simple app shows only some of the capabilities of the Custom Vision service. To learn more about what you can do with this service, see the [Custom Vision page](#).