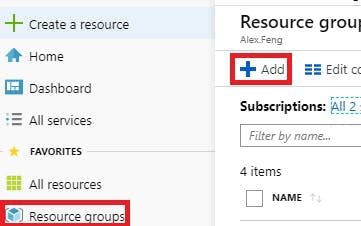


**Azure Resource Group setup**

Go to Azure portal [https://portal.azure.com](https://portal.azure.com/)

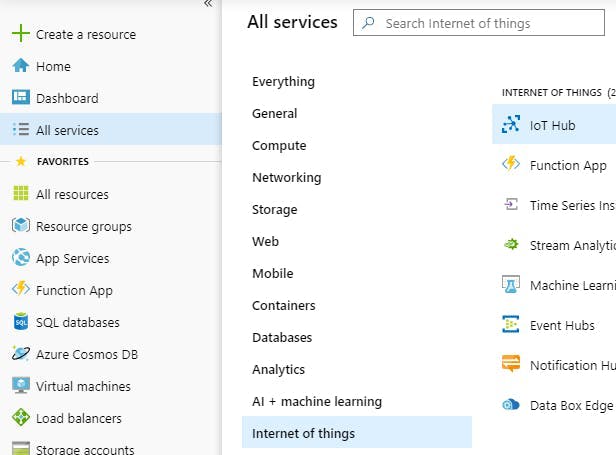
An Azure Resource Group is basically a folder where are your resources will be located for this project. Create one as shown below and name it whatever you'd like.



* I will refer the name of your resource group as <RESOURCE GROUP>

**Azure IoT setup**

1. Create an "IoT hub"



* For "Resource group", select the one you just created
* I will refer the name of your hub as <IOT HUB NAME>

The Particle serves as a a light dimmer. Since it is hard to distinct the various brightness levels on the PWM led, I chose to use 3 leds to represent 0, 25, 50, 75, & 100% brightness.

When 24 hours of data has been collected, the Particle sends it to the Azure integration, which I'll show how to setup here.

***Setup the Azure Integration:***

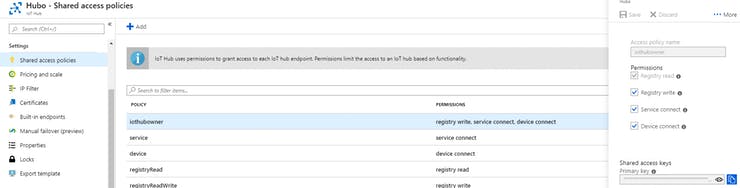
* *Go* to <https://console.particle.io/integrations/azure-iot-hub/create>

- Event Name = dataSetFull

- IoT Hub Name = <IOT HUB NAME>

- Shared Policy Name = iothubowner

- Get your "Shared Policy Key" by opening up your IoT Hub and doing this:



***Setup the Machine Learning Prediction Integration*** ***(complete the Azure machine learning setup before*** ***proceeding):***

* Go to <https://console.particle.io/integrations/webhooks/create>

- Event Name = getPrediction

- URL = <REQUEST URL>

- Request Type = POST

- Request Format = JSON

* Expand "Advanced Settings"
* In JSON Data, select "custom" & paste this

{

"Inputs": {

"input1": [

{

"Hour": "{{{PARTICLE\_EVENT\_VALUE}}}"

}

]

}

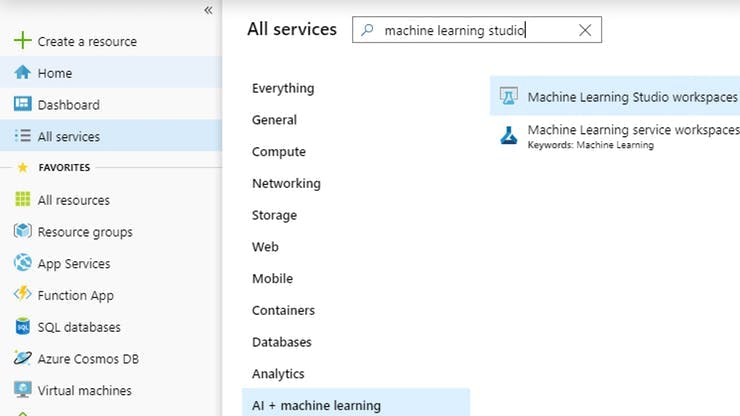
}

- Note: leave PARTICLE\_EVENT\_VALUE as it is

* In HTTP Headers, add "Authorization" > "Bearer <PREDICT KEY>"

**Azure Machine Learning setup**

1. Create a "machine learning studio workspace"



* select your existing resource group
* choose create a new storage - this will hold your datasets and machine learning models - I will refer this as <STORAGE NAME>
* For location, choose "South Central US". Certain regions don't have the "Machine Learning Studio" service available.

2. Copy the training experiment: <https://gallery.azure.ai/Experiment/Smart-Home-Train>. This is what trains your models.

3. When the experiment has opened up, at the bottom menu bar, click "Run". When finished, hover over "Set Up Web Service" and select "Deploy Web Service [New] Preview".

4. Copy the predictive experiment: <https://gallery.azure.ai/Experiment/Smart-Home-Predict>. This is what the Particle device uses to determine the brightness of the light depending on the hour of the day. Run & deploy this as a new web service as well.

* I will refer the name of your predictive web service as <PREDICTIVE WEB SERVICE NAME>

5. Now you need to record the REST API info which will be used later when creating your Azure Logic App.

* Go to <https://services.azureml.net/webservices>
* Click the link to your "Smart Home Train" web service & click "Consume" (located at the top menu bar)

- Copy "Primary Key" - I will refer this as <TRAIN KEY>

- Copy "Batch Requests" - I will refer this as <TRAIN URL>

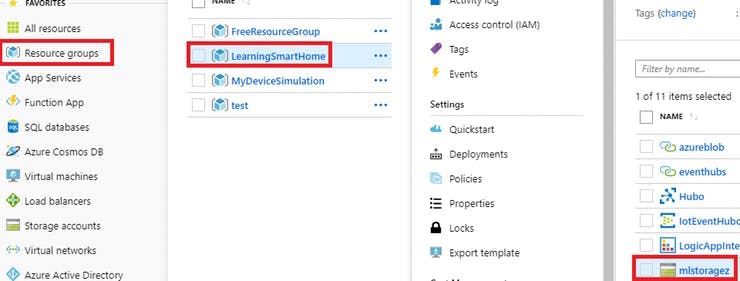
* Now open up your "Smart Home Predict" web service

- Copy "Primary Key" - I will refer this as <PREDICT KEY>

- Copy "Request-Response" - I will refer this as <PREDICT URL>

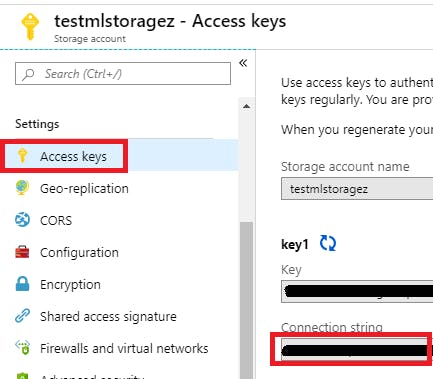
**Azure Storage setup**

1. Open your storage (created from the machine learning studio setup)



2. Scroll down to "Access Keys" in the left menu

* copy either key1 or key2 "Connection String" - I will refer this as <STORAGE CONNECT STRING>



3. Scroll down and click "Blobs" in the left menu

4. Click "+ Container" to add a new container (this will be used when we create our logic app) - I will refer this as <STORAGE CONTAINER>

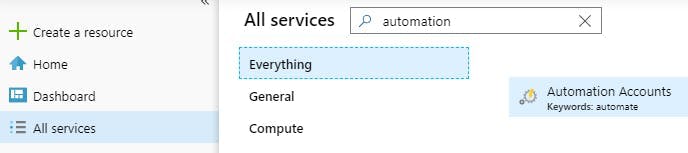
5. Open the container & upload "data.csv" from the same github repo. Keep the default name.

* <https://github.com/gearsmotion789/LearningSmartHomeParticle>

**Azure Automation setup**

Automations can run Powershell commands automatically. They are used to update our predictive web service to reference the new, retrained model. It will be used the logic app we'll create in the next section.

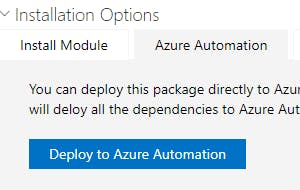
1. Create an "Automation Accounts"



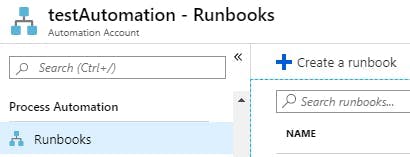
* I will refer the name of your automation as <AUTOMATION NAME>

2. We need to install the Az.MachineLearning cmdlets for Powershell at [https://www.powershellgallery.com/packages/Az.MachineLearning/1.1.0](https://www.powershellgallery.com/packages/Az.MachineLearning/1.1.03)

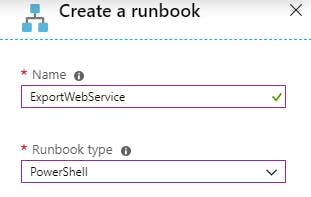
* select the "Azure Automation" tab under "Installation Options" & Deploy



3. Open your "automation account" and create a "runbook", called "ExportWebService"



* It should look like this



4. Click "edit" & paste "ExportWebService.ps1" (located in the code section)

* Replace <YOUR AZURE EMAIL>
* Replace <YOUR AZURE PASSWORD>
* Replace <TENANT ID> - copy the "Directory ID" from <https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/Properties>
* Replace <SUBSCRIPTION ID> - copy the "Subscription ID" from <https://portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade>
* Replace <PREDICTIVE WEB SERVICE NAME>
* Replace <RESOURCE GROUP>
* Click "save" & then "publish"

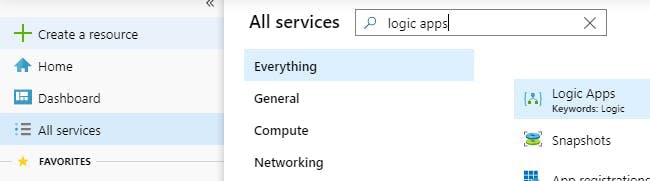
5. Create another "runbook", called "ImportWebService" & paste "ImportWebService.ps1" (located in the code section) - save & publish it

**Azure Logic App setup**

The way this logic app works is:

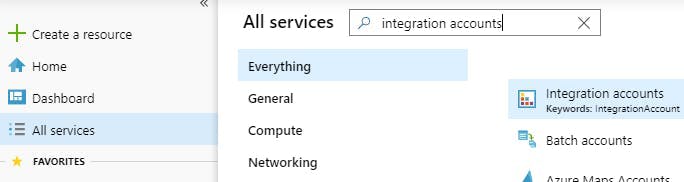
* when we receive new data from the device, append that data with the old data
* convert the data into csv file format
* upload & replace the old csv dataset with the new one in the machine learning storage (created in the previous section)
* create a retraining job, referencing the location of the csv dataset
* start the retraining job
* update the predictive web service to reference the new, retrained model

1. In the Azure portal, create a "Logic Apps"



* For "Resource Group", select your already existing one
* For location, choose "East US". For some reason the logic app has issues in "South Central US".

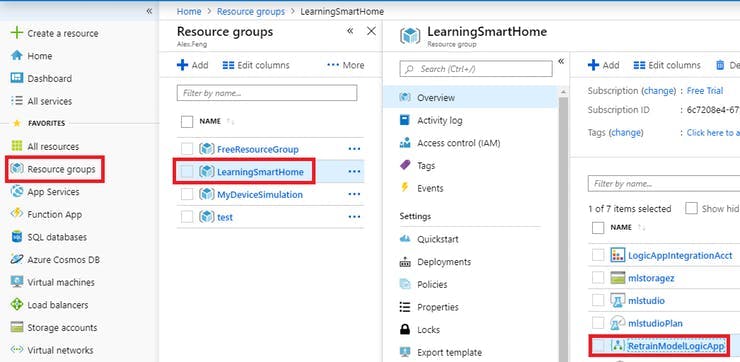
2. Also create an "Integration Accounts" - this allows for inline javascript code



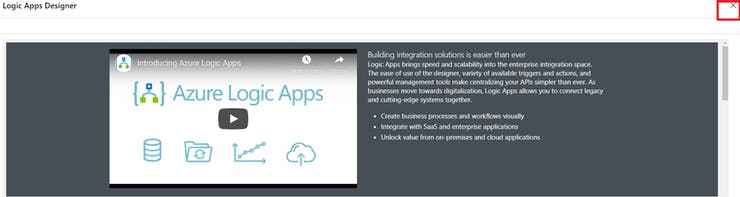
3. One feature in the logic app depends on an external service, "Plumbsail - Parse csv". So you need to create a "Plumbsail **Documents**" api key at <https://account.plumsail.com/documents/api-keys>. Note: you don't have to register your Plumbsail account with the same email as your Azure account.

* I will refer this as <PLUMBSAIL KEY>

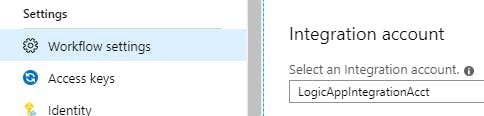
4. Now open your logic app



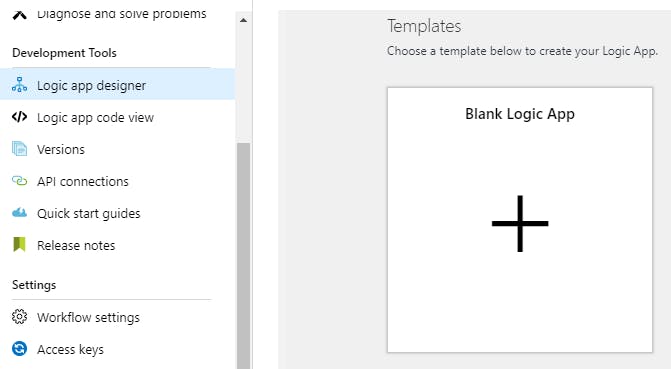
* If you see a welcome screen like this, just click the "X" to close it



5. Link the "Integration account" you just created & "save"

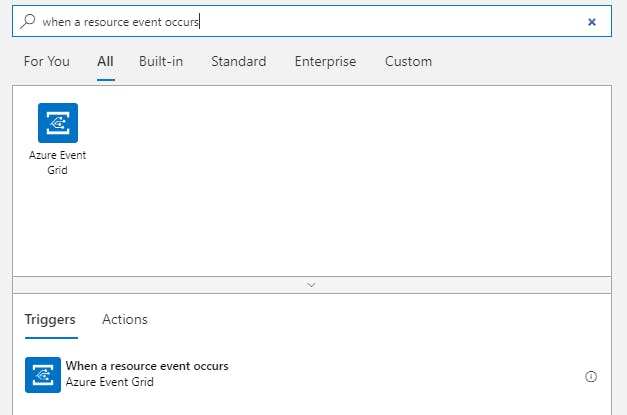


6. Click on "Logic App Designer" & create a new logic app by choosing "Blank Logic App" under "Templates"

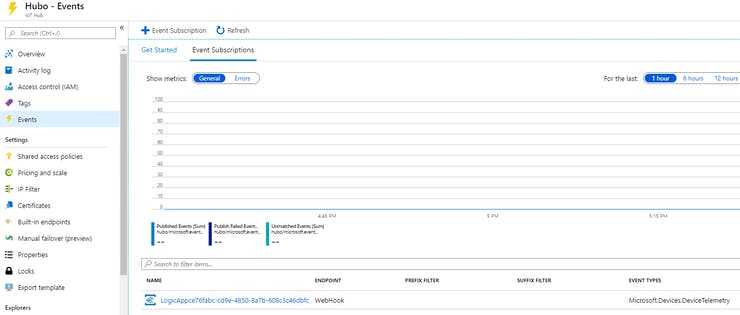


7. Now you need to setup the connections to all of your resources.

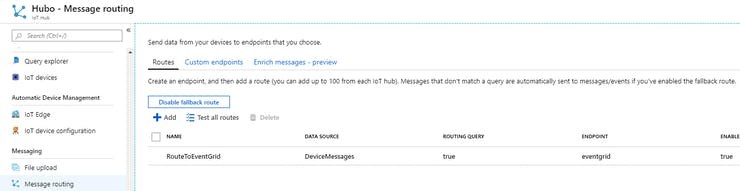
* Search for "when a resource event occurs", click on the trigger, and click "Sign In"



* After signing in, don't fill out the details. Click the "+ Next Step" button below the tile
* Search for "get blob content". Enter a connection name, and choose your storage. Click "Create". Don't fill out the details.
* Click "+ Next Step". Search for "parse csv". Enter your <PLUMSAIL KEY>. Click "Create".
* Click "+ Next Step". Search for "create job". Click "Sign In".
* ***Warning:****Device telemetry via the Event Grid is still in "Preview", so it may not work. If you don't see an Event Subscription to your Logic App in the Iot Hub resource, it means the link between the services was unsuccessful.*



* *Also check if a "Messaging Route" to "eventgrid" has appeared. This means device telemetry messages can successfully transfer to the Event* Grid*, and therefore to your logic app.*

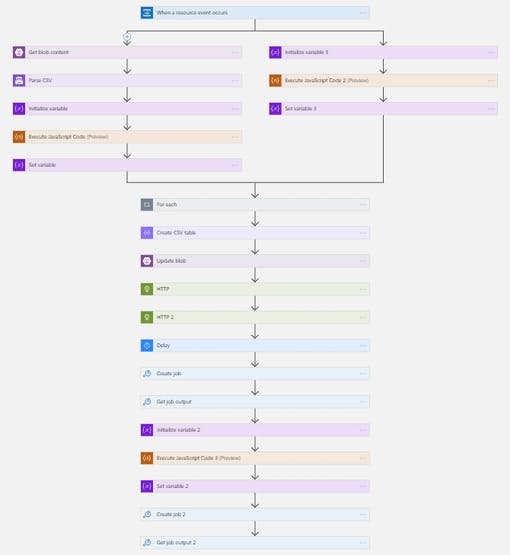


8. Click the "Code View" tab

9. Paste "LogicApp.json" (located in the code section) - replace all the details listed below. Note: you can use a software like "Sublime Text" (<https://www.sublimetext.com/download>) & do "Ctrl H" to "replace all".

* Replace <LOCATION> (e.g. eastus)
* Replace <YOUR SUBSCRIPTION ID>
* Replace <RESOURCE GROUP>
* Replace <STORAGE NAME>
* Replace <STORAGE CONTAINER>
* Replace <STORAGE CONNECT STRING>
* Replace <AUTOMATION NAME>
* Replace <TRAIN KEY>
* Replace <TRAIN URL>
* Replace <TRAIN URL 2> - take your <TRAIN URL> and remove the url query, "?api-version=2.0"
* Replace <IOT HUB NAME>

10. After saving, to check everything is setup correctly, click on the "Designer" tab. Your workflow should look like this:



11. Click on the "Get Blob Content" tile, and replace <STORAGE FILE> with the location to "data.csv" in your storage container



12. Repeat step 11, but for "Update Blob" as well