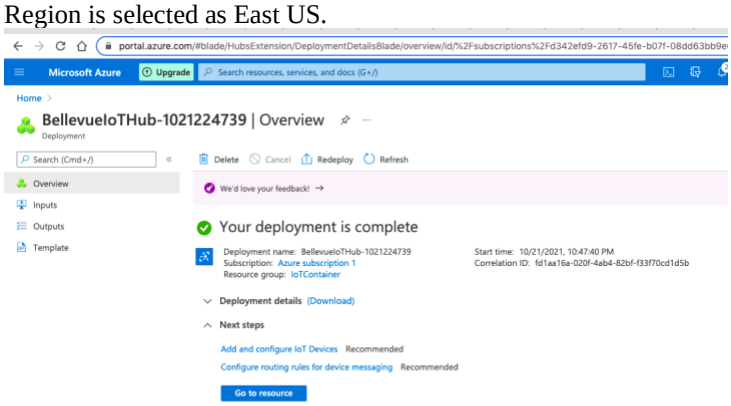
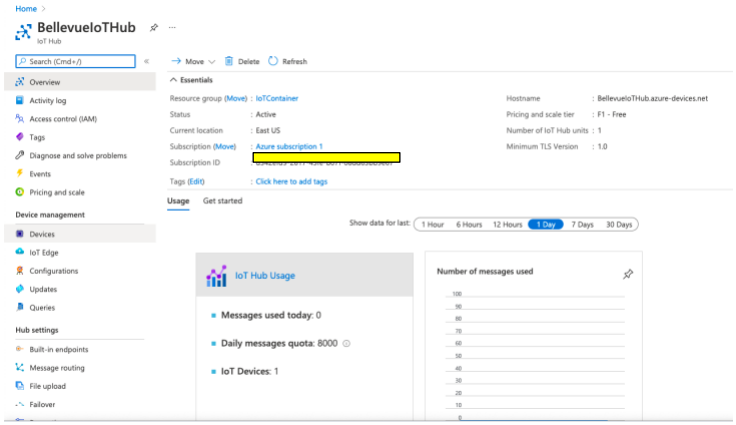
Homework – 2

Sayali Kudale

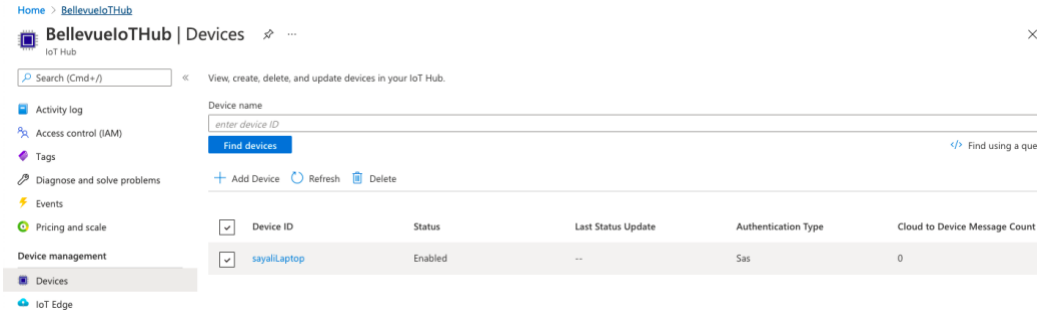
Part 1 : Configure laptop as IoT Device

1. Created Azure account.
2. Created an IoT Hub.
    a. An IoT hub with name "BellevueIoTHub" created using azure portal. In this example we have selected the F1- free pricing plan.
    b. A new Resource group is created with name IoTContainer.
    c. Region is selected as East US.

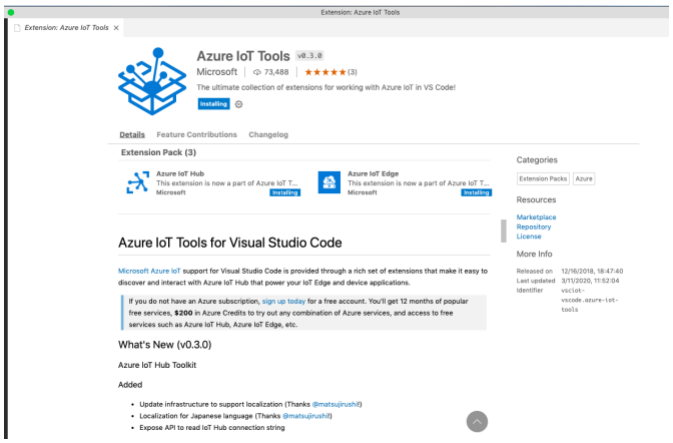




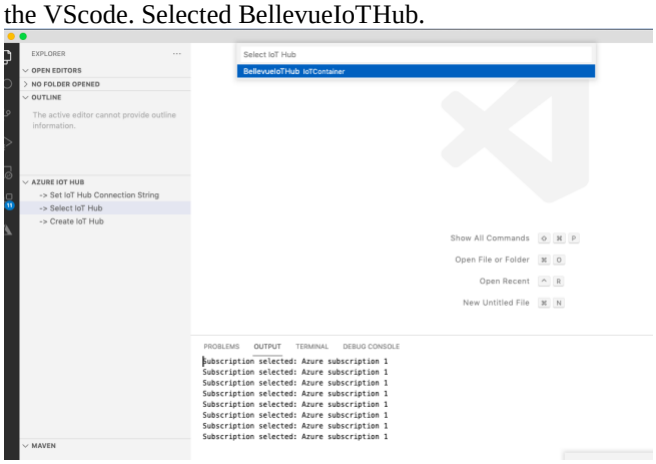
3. Created Device
    a. Inside BellevueIoTHub we have added new device with name 'sayaliLaptop'



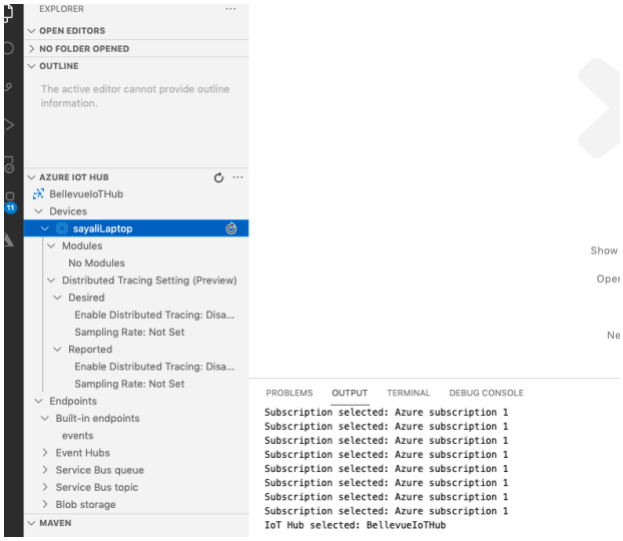
4. Configure the vscode for Azure functionalities:
    a. Added Azure IoT tools extensions in VsCode.



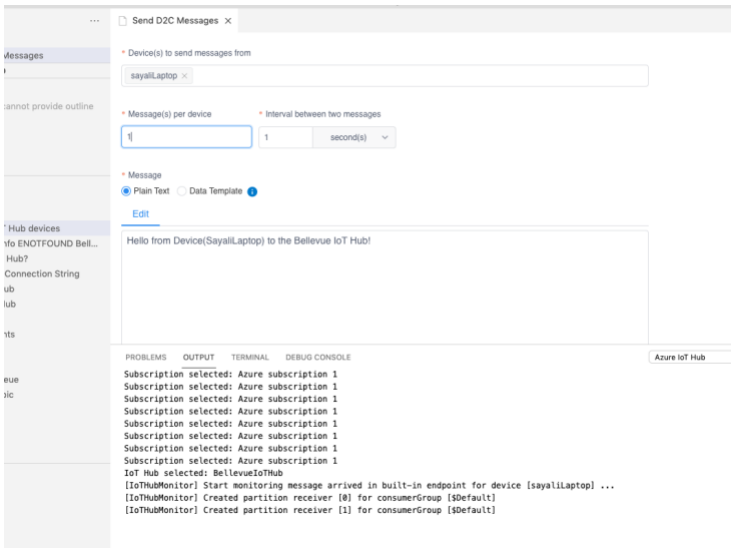
b. In VsCode added azure subscription information and we could see the created IoT hub in step 1 in the VScode. Selected BellevueIoTHub.



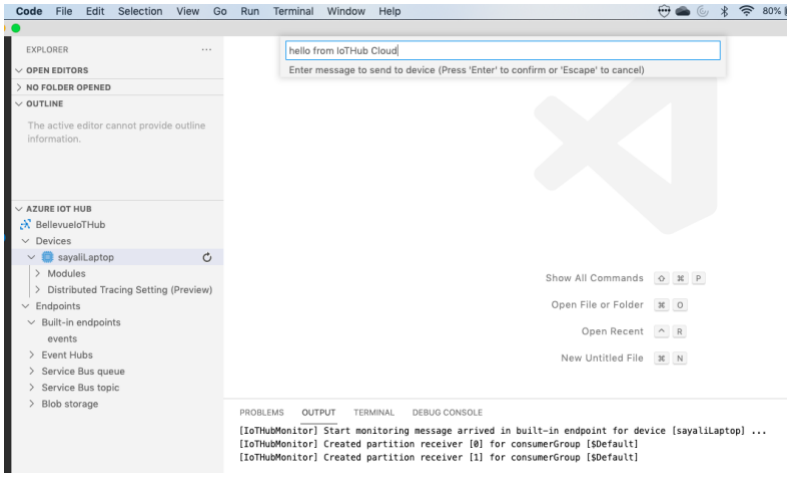
c. We could also see the device information in the VScode.



5. Send messages from Vscode.
   a. On right click of the device ‘sayaliLaptop’, we get the options to send messages from device to cloud and vice versa.
   b. With the help of ‘send D2C messages’ we can send messages from device to the cloud.



c. With the help of 'send C2D messages' we can send messages from configured IoT hub to the device.



Part 2 : Configure IoT hub to receive messages from device

1. Installed Azure CLI on macOs with the help of homebrew commands.
2. On the terminal login into the azure. Please find below the terminal output after login into azure:

   (base) Abhijeets-MacBook-Air:~ sayali$ az login
   The default web browser has been opened at
   https://login.microsoftonline.com/common/oauth2/authorize. Please continue the login in the web

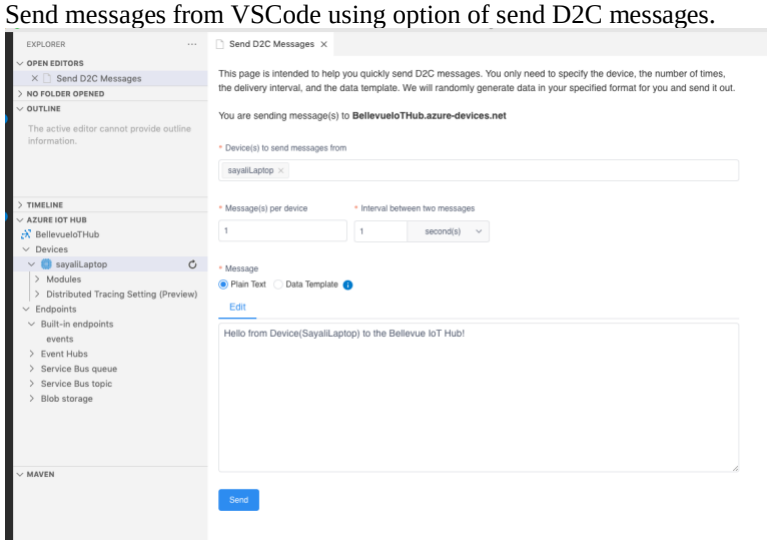
browser. If no web browser is available or if the web browser fails to open, use device code flow with `az login --use-device-code`.
You have logged in. Now let us find all the subscriptions to which you have access...


```
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "**************",
    "id": "**************",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure subscription 1",
    "state": "Enabled",
    "tenantId": "**************",
    "user": {
      "name": "*******@gmail.com",
      "type": "user"
    }
  }
]
```


3. Verify login into the azure IoT hub by viewing the device connection string for hub BellevueIoTHub and Device sayaliLaptop.

```
(base) Abhijeets-MacBook-Air:~ sayali$ az iot hub device-identity connection-string show --device-id sayaliLaptop --hub-name BellevueIoTHub
{
  "connectionString": "HostName=BellevueIoTHub.azure-devices.net;DeviceId=sayaliLaptop;SharedAccessKey=********************************* "
}
```

4. Send messages from VSCode using option of send D2C messages.

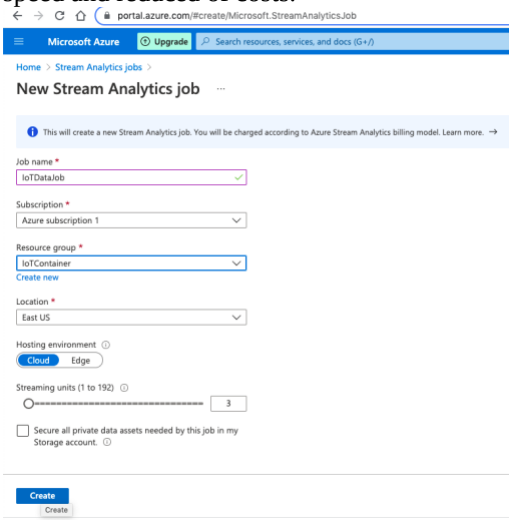


5. Monitor the messages of Azure IoT hub in terminal using az iot hub monitor-events commands.

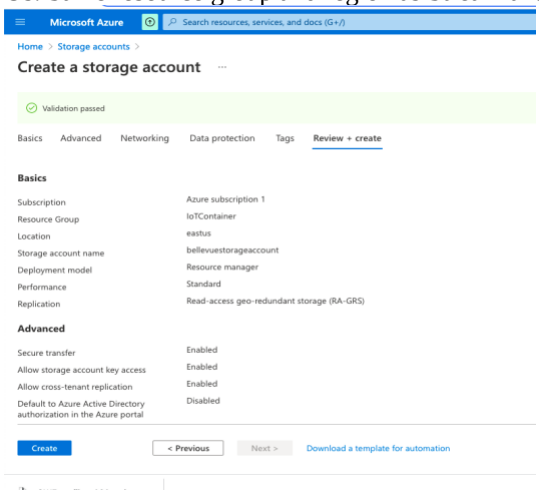
```
(base) Abhijeets-MacBook-Air:~ sayali$ az iot hub monitor-events --hub-name BellevueIoTHub --output json
Starting event monitor, use ctrl-c to stop...
{
    "event": {
        "origin": "sayaliLaptop",
        "module": "",
        "interface": "",
        "component": "",
        "payload": "Hello from Device(SayaliLaptop) to the Bellevue IoT Hub!"
    }
}
```

Part 3 : Connect Azure Stream Analytics

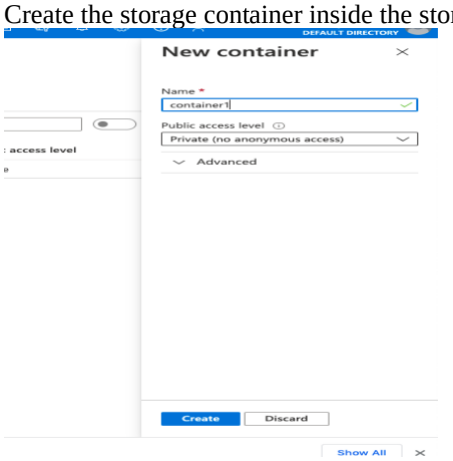
1. Created a new Stream analytics job for the resource group IoTContainer, which is same resource group of BellevueIoTHub. We have used same location of resource group and all resources to increased processing speed and reduced of costs.



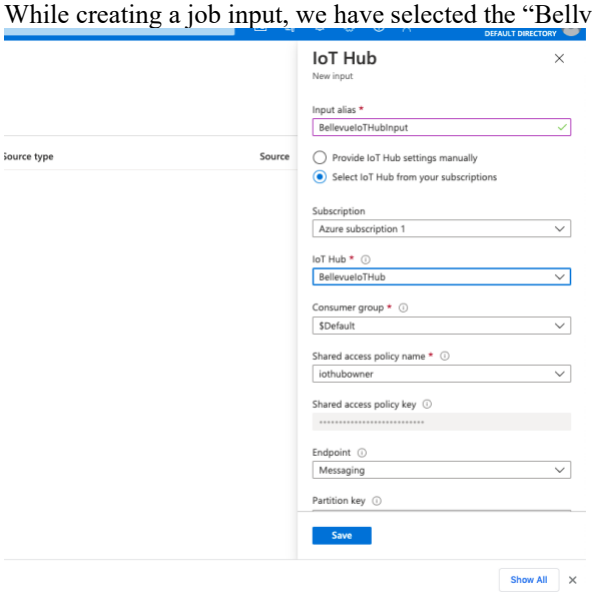
2. Create a storage account.
   a. We have created a storage account for the "IoTContainer" resource group and region is also East Us. Same resource group and region as Stream analytics job and IoT hub.



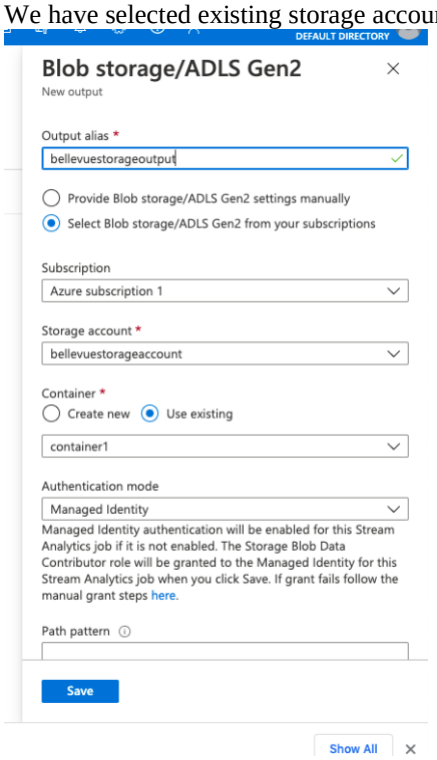
   b. Create the storage container inside the storage account with the name 'container1'.



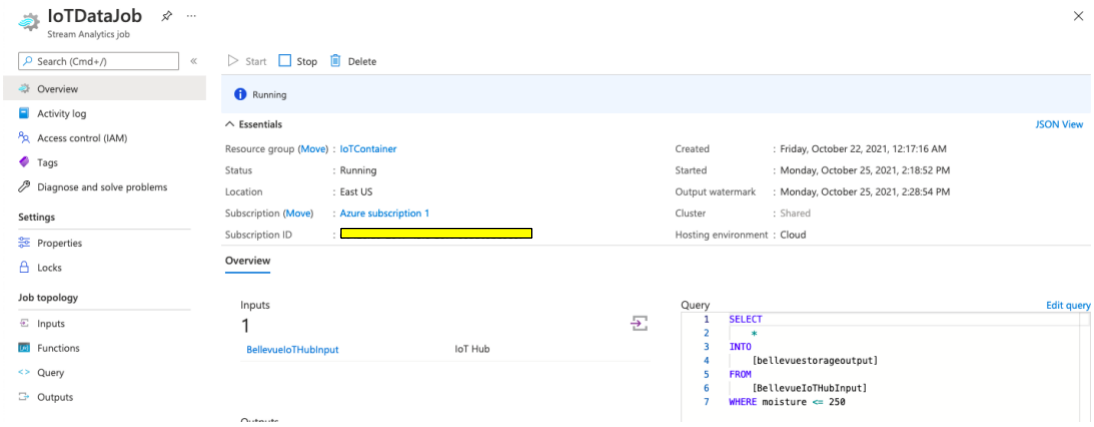
3. For Stream analytics job we need to create input, output and Query.
4. Configure job input:
    a. We can configure an IoT hub device input to the stream analytics job.
    b. While creating a job input, we have selected the “BellvueIoTHub” as IoT hub.



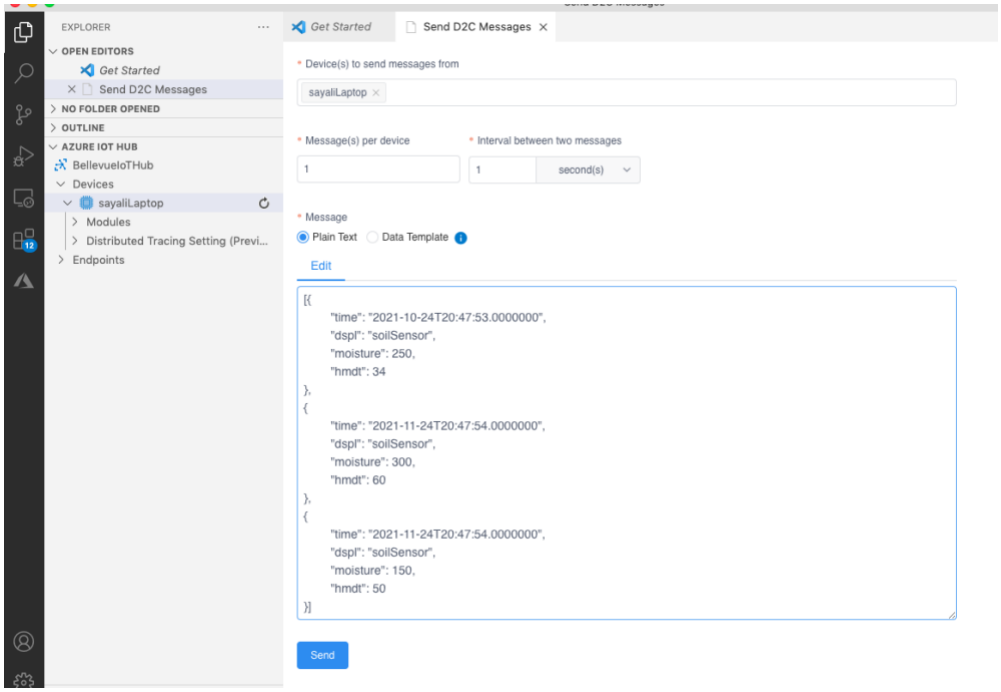
5. Configure job output:
    a. We have created new output for the stream analytics job with name ‘bellevueStorageOutput’’.
    b. We have selected existing storage accounts and container created in step 2.



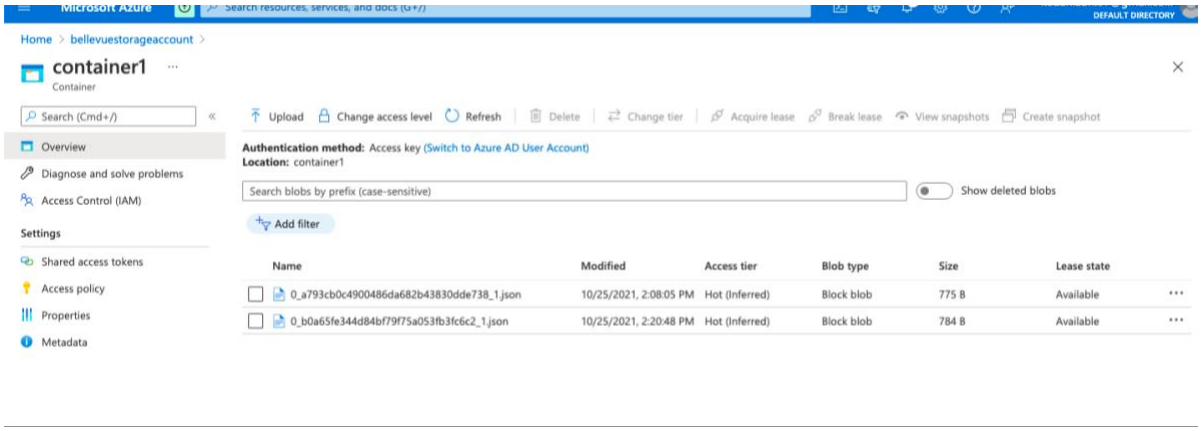
6. Update the query of stream analytics job. In this example, we have updated query such that only those inputs will be saved to “bellevueStorageoutput” whose moisture value is less than or equal to 250.



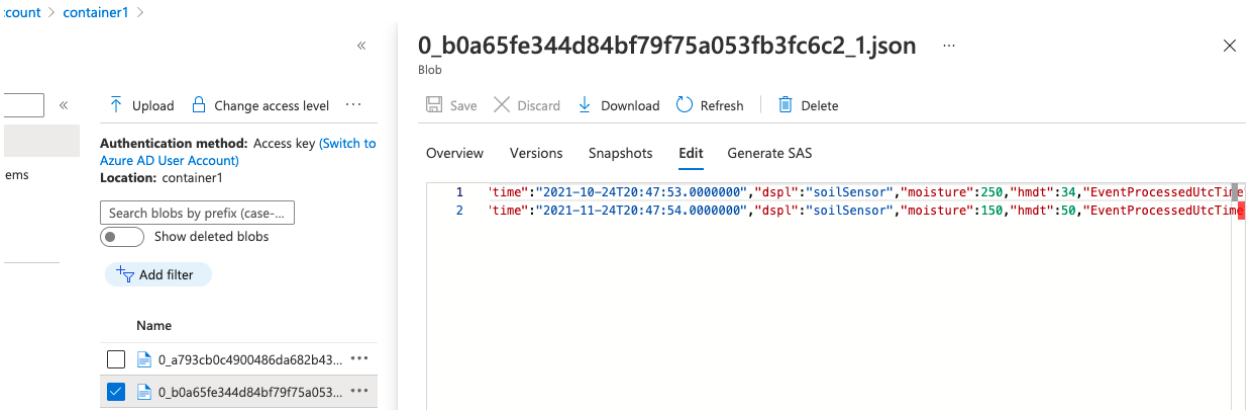
7. Start the stream analytics job and check the output.
    a. We have started the stream analytics job. And now it is ready to accept the input messages.
    b. From VSCode , send the data in json format. We have send a sample data with moisture value 250, 300 and 150.



    c. As stream analytics job was running, it accepted the input and based on the query it saved the result in output which is, bellevuestorageaccount container1.



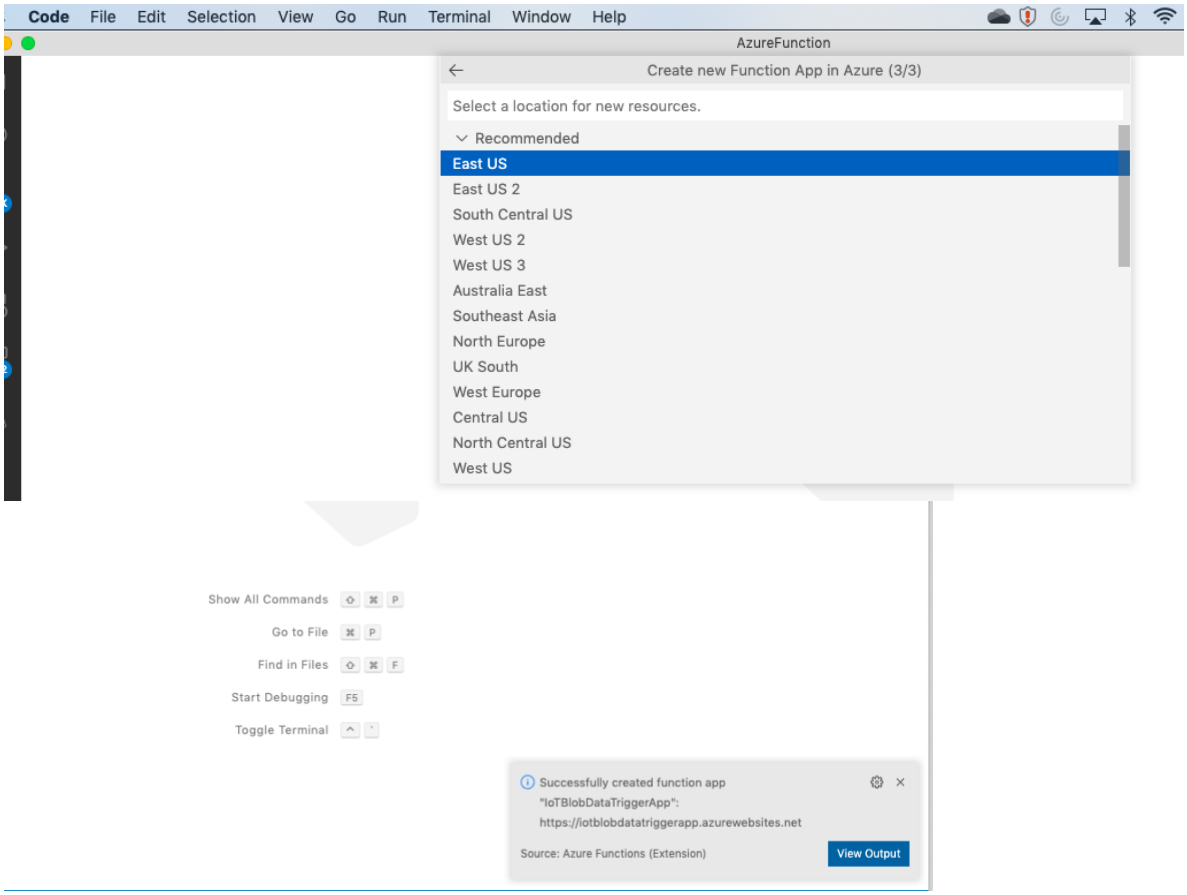
d. The json file contains only 2 records of sensor data with moisture value 150, 250. The moisture value 300 data is not present in the data because as per the job query moisture should be less than or equal to 250.

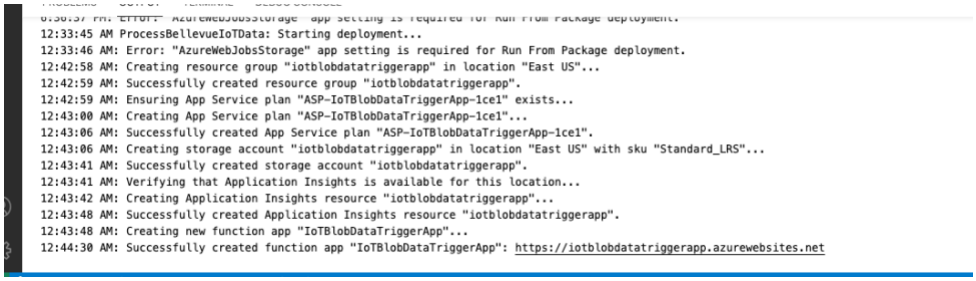


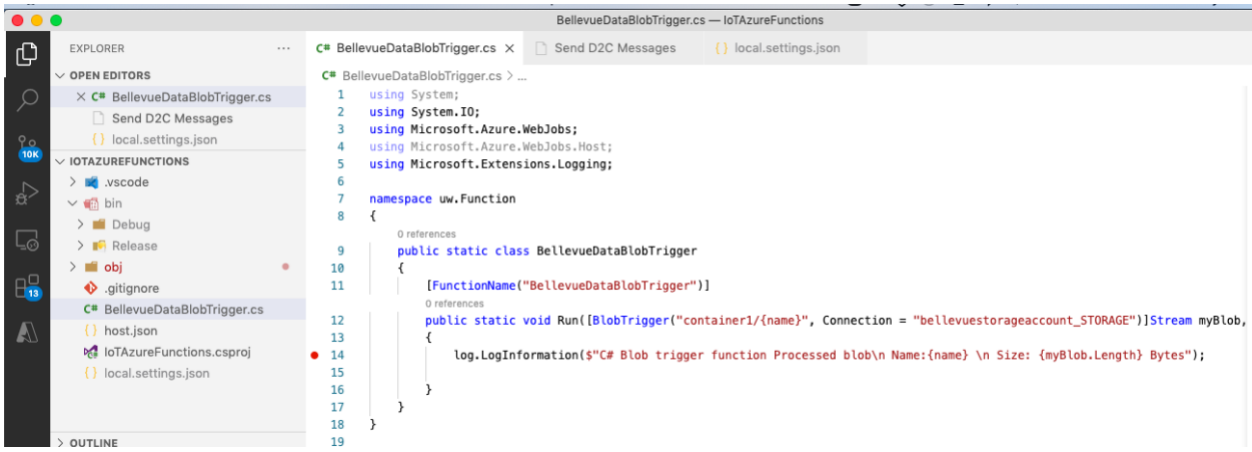
4. Connect Azure Functions to respond to Azure Blob events

In this part we have created azure function app that triggers whenever new blob is uploaded. Below steps has been followed:

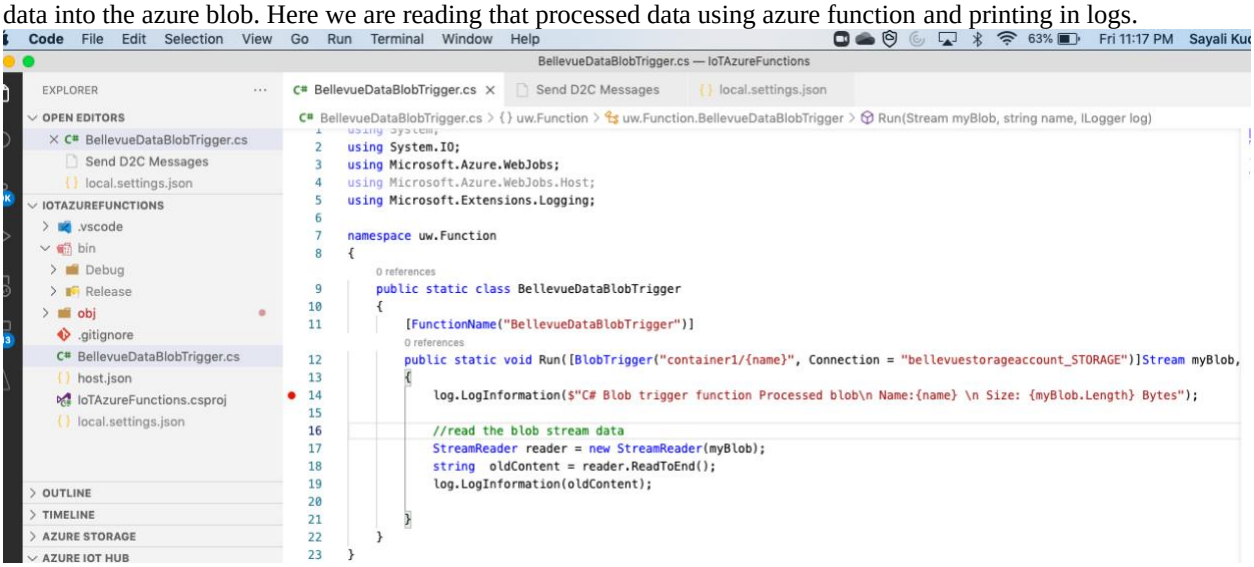
1. Install extensions in vsCode, to configure it to develop Azure functions locally:
    a. Installed Azure functions core tools extensions in vscode.
    b. Installed C# extensions in vscode.
    c. Installed dot net core cli tools.
    d. Installed Azure functions extensions in vscode.
2. Create Azure function project in vscode.
    a. While creating project we have selected Azure blob storage trigger as function template and function app name as IoTBlobDataTriggerApp.
    b. We have selected region as East US, which is being used by stream analytics job and storage account in part 3.
    c. Selected azure function name as “BellevueDataBlobTrigger” and namespace as uw.Function.
    d. While selecting local.settings.json, we have selected “bellevuestorageaccount_Storage.” This is a setting of blob storage account that is being used by the stream analytic job in part 3.
    e. We have specified path as ‘container1’, this is the container inside bellevuestorageaccount ans being used by the stream analytic job in part 3.
    f. After this function template has been created in visual studio code.


Code File Edit Selection View Go Run Terminal Window Help
AzureFunction
Create new Function App in Azure (3/3)
Select a location for new resources.
Recommended
East US
East US 2
South Central US
West US 2
West US 3
Australia East
Southeast Asia
North Europe
UK South
West Europe
Central US
North Central US
West US
Show All Commands
Go to File
Find in Files
Start Debugging
Toggle Terminal
Successfully created function app "IoTBlobDataTriggerApp": https://iotblobdatatriggerapp.azurewebsites.net
Source: Azure Functions (Extension)
View Output

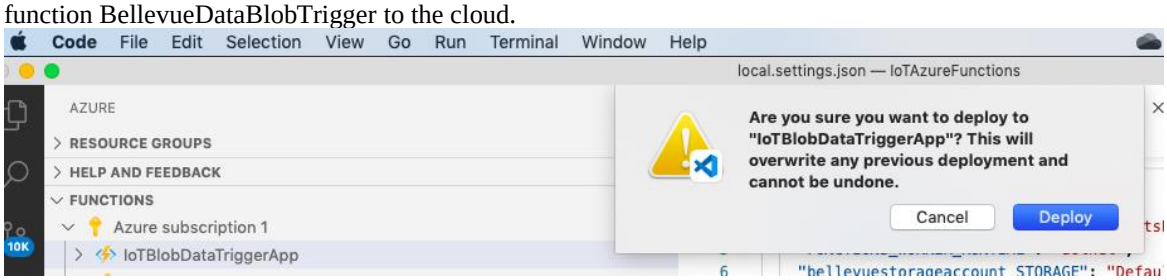

12:33:45 AM ProcessBellevueIoTData: Starting deployment...
12:33:46 AM: Error: "AzureWebJobsStorage" app setting is required for Run From Package deployment.
12:42:58 AM: Creating resource group "iotblobdatatriggerapp" in location "East US"...
12:42:59 AM: Successfully created resource group "iotblobdatatriggerapp".
12:42:59 AM: Ensuring App Service plan "ASP-IoTBlobDataTriggerApp-1ce1" exists...
12:43:00 AM: Creating App Service plan "ASP-IoTBlobDataTriggerApp-1ce1"...
12:43:06 AM: Successfully created App Service plan "ASP-IoTBlobDataTriggerApp-1ce1".
12:43:06 AM: Creating storage account "iotblobdatatriggerapp" in location "East US" with sku "Standard_LRS"...
12:43:41 AM: Successfully created storage account "iotblobdatatriggerapp".
12:43:41 AM: Verifying that Application Insights is available for this location...
12:43:42 AM: Creating Application Insights resource "iotblobdatatriggerapp"...
12:43:48 AM: Successfully created Application Insights resource "iotblobdatatriggerapp".
12:43:48 AM: Creating new function app "IoTBlobDataTriggerApp"...
12:44:30 AM: Successfully created function app "IoTBlobDataTriggerApp": https://iotblobdatatriggerapp.azurewebsites.net

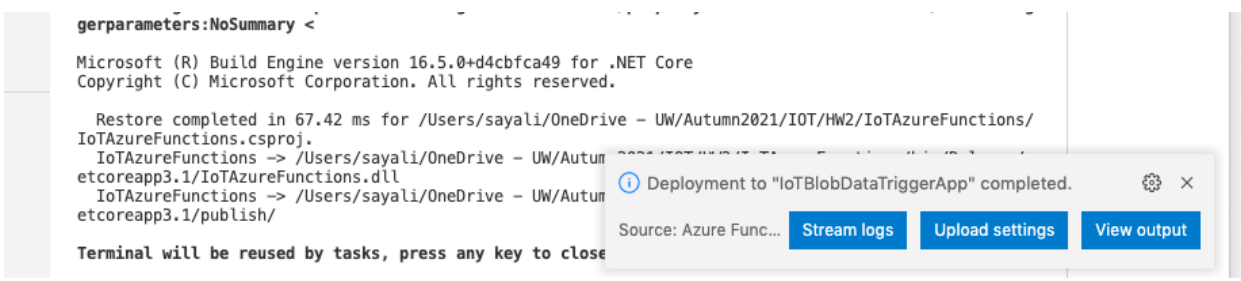

BellevueDataBlobTrigger.cs — IoTAzureFunctions
EXPLORER
OPEN EDITORS
BellevueDataBlobTrigger.cs
Send D2C Messages
local.settings.json
IOTAZUREFUNCTIONS
.vscode
bin
Debug
Release
obj
.gitignore
BellevueDataBlobTrigger.cs
host.json
IoTAzureFunctions.csproj
local.settings.json
OUTLINE
using System;
using System.IO;
using Microsoft.Azure.WebJobs;
using Microsoft.Azure.WebJobs.Host;
using Microsoft.Extensions.Logging;
namespace uw.Function
{
0 references
public static class BellevueDataBlobTrigger
{
[FunctionName("BellevueDataBlobTrigger")]
0 references
public static void Run([BlobTrigger("container1/{name}", Connection = "bellevuestorageaccount_STORAGE")]Stream myBlob,
{
log.LogInformation($"C# Blob trigger function Processed blob\n Name:{name} \n Size: {myBlob.Length} Bytes");
}
}
}

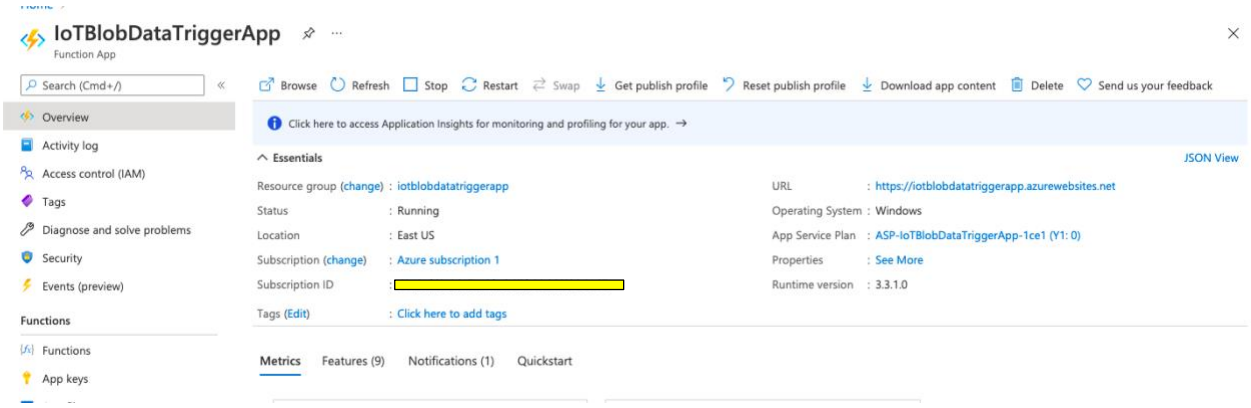
3. Local.settings.json in the generated project contains the connection strings of the webJobsStorage and blob storage account, with the key name as AzureWebJobsStorage and bellevuestorageaccount_STORAGE. bellevuestorageaccount_STORAGE key is used as value for connection attribute in azure function that is generated.
4. Modified the Azure function to read the blob stream data. In part 3, stream analytics job saved the processed data into the azure blob. Here we are reading that processed data using azure function and printing in logs.



5. Publish the azure function to cloud:
   a. Build the azure function in vscode.
   b. Using publish option in vscode editor deploy the IoTBlobDataTriggerApp function app and azure function BellevueDataBlobTrigger to the cloud.



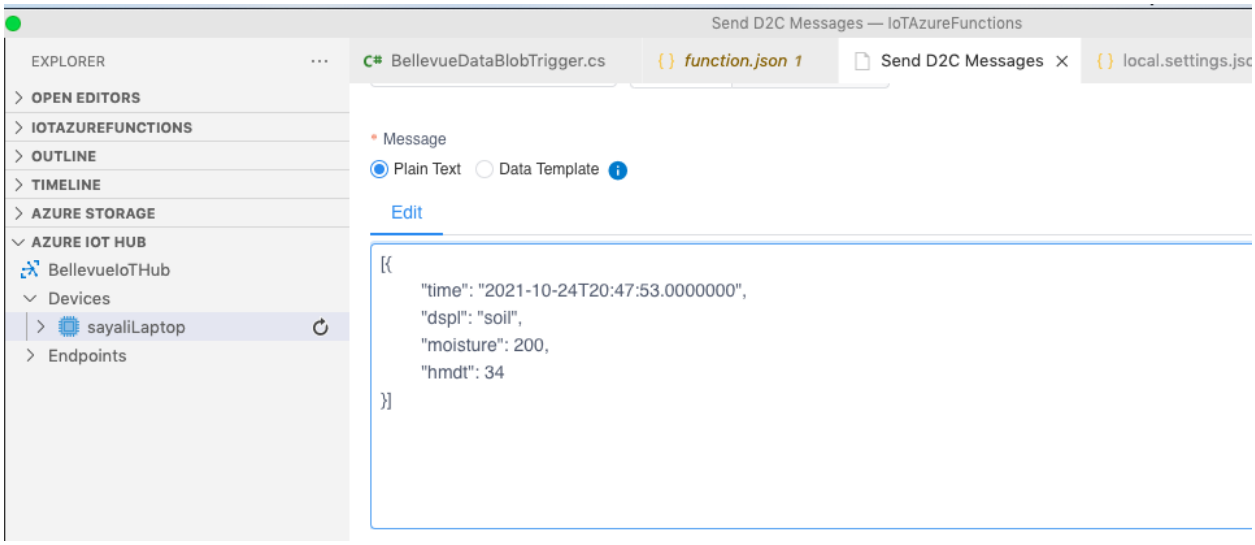




6. Add connection string in the portal. New setting with key as bellevuestorageaccount_STORAGE and value as connection string of BellevueStorageAccount.

Name	Value	Source	Deployment slot setting	Delete	Edit
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click to show value	App Service Config			
AzureWebJobsStorage	DefaultEndpointsProtocol=https;Accoun	App Service Config			
bellevuestorageaccount_STORAGE	Hidden value. Click to show value	App Service Config			
FUNCTIONS_EXTENSION_VERSION	Hidden value. Click to show value	App Service Config			
FUNCTIONS_WORKER_RUNTIME	Hidden value. Click to show value	App Service Config			
WEBSITE_CONTENTAZUREFILECONNECTION:	Hidden value. Click to show value	App Service Config			
WEBSITE_CONTENTSHARE	Hidden value. Click to show value	App Service Config			
WEBSITE_RUN_FROM_PACKAGE	Hidden value. Click to show value	App Service Config			

7. Test the function locally and in cloud:
   a. Run the stream analytics job and newly created web app function.
   b. From device "sayaliLaptop" send D2C message in the json format. Stream analytics job created in step 3 will process this message and add new blob file to storage account(bellevuestorageaccount) and container1.
   c. As new file is added in the blob storage, this "BellevueDataBlobTrigger" function triggers and prints the content of the file in terminal.
   d. If we click monitor stream data in vscode terminal,we can monitor logs.
   e. Logs can also be monitored in the portal in the log stream tab.
   f. In the azure portal we can also view the graphs of memory working set and function execution count.



Home > bellevuestorageaccount >

0_08466c940919484198cba35dfa0ebcb0_1.json

Blob

Save | Discard | Download | Refresh | Delete

Overview | Versions | Snapshots | Edit | Generate SAS


```
1  {"time":"2021-10-24T20:47:53.0000000","dspl":"soil","moisture":200,"hmdt":34,"EventProcessedUtcTime":"2021-10-30T05:02:49.5201320Z","PartitionId":0,"EventEnqueuedUtcTime":"2021-10-30T0
```


Send

PROBLEMS 8 | OUTPUT | TERMINAL | DEBUG CONSOLE | host start - Task - netcoreapp3.1

```
 file configuration.
[2021-10-30T05:10:16.421Z] Cannot create directory for shared memory usage: /dev/shm/AzureFunctions
[2021-10-30T05:10:16.421Z] System.IO.FileSystem: Access to the path '/dev/shm/AzureFunctions' is denied. Operation not permitted.

Functions:

        BellevueDataBlobTrigger: blobTrigger

For detailed output, run func with --verbose flag.
[2021-10-30T05:10:24.042Z] Executing 'BellevueDataBlobTrigger' (Reason='New blob detected: container1/0_08466c940919484198cba35dfa0ebcb0_1.json',
Id=1f2724d3-4ab8-42a9-8d54-3229d73504fe)
[2021-10-30T05:10:24.105Z] C# Blob trigger function Processed blob
[2021-10-30T05:10:24.106Z]  Name:0_08466c940919484198cba35dfa0ebcb0_1.json
[2021-10-30T05:10:24.106Z]  Size: 385 Bytes
[2021-10-30T05:10:24.210Z] {"time":"2021-10-24T20:47:53.0000000","dspl":"soil","moisture":200,"hmdt":34,"EventProcessedUtcTime":"2021-10-30T05:02:
49.5201320Z","PartitionId":0,"EventEnqueuedUtcTime":"2021-10-30T05:02:49.3280000Z","IoTHub":{"MessageId":null,"CorrelationId":null,"ConnectionDevi
ceId":"sayaliLaptop","ConnectionDeviceGenerationId":"637704791389386368","EnqueuedTime":"2021-10-30T05:02:48.8500000Z"}}
[2021-10-30T05:10:24.240Z] Executed 'BellevueDataBlobTrigger' (Succeeded, Id=1f2724d3-4ab8-42a9-8d54-3229d73504fe, Duration=419ms)
[2021-10-30T05:10:25.108Z] Host lock lease acquired by instance ID '000000000000000000000000FC3D60D8'.
[2021-10-30T05:10:25.199Z] Assembly reference changes detected. Restarting host...

Terminal will be reused by tasks, press any key to close it.
```

Home > IoTBlobDataTriggerApp

IoTBlobDataTriggerApp | Log stream

Function App

Search (Cmd+/)

CORS

Monitoring

- Alerts
- Metrics
- Health check
- Logs
- Diagnostic settings
- App Service logs
- Log stream
- Process explorer

Filesystem Logs | Log Level | Stop | Copy | Clear | Leave Feedback

```
Connected!
2021-10-30T05:28:31  Welcome, you are now connected to log-streaming service. The default timeout is 2 hours. Change the timeout
with the App Setting SCM_LOGSTREAM_TIMEOUT (in seconds).
2021-10-30T05:29:25.926 [Information] Executing 'BellevueDataBlobTrigger' (Reason='New blob detected:
container1/0_08466c940919484198cba35dfa0ebcb0_1.json', Id=e4c9c83f-d4ef-4cd7-a385-496c401b8700)
2021-10-30T05:29:25.927 [Information] C# Blob trigger function Processed blobName:0_08466c940919484198cba35dfa0ebcb0_1.jsonSize:
772 Bytes
2021-10-30T05:29:25.927 [Information] Executed 'BellevueDataBlobTrigger' (Succeeded, Id=e4c9c83f-d4ef-4cd7-a385-496c401b8700,
Duration=6ms)
2021-10-30T05:29:28.158 [Information] Host Status: {"id": "iotblobdatatriggerapp","state": "Running","version":
"3.3.1.0","versionDetails": "3.3.1 Commit hash: bd2b318013cee8d608c88203ec4ee3516d2a9961","platformVersion":
"95.0.7.554","instanceId": "d141dfc91ce38ea7c837f8110af3fb3664d428da4dac25484a7350a5290960d9","computerName": "10-30-5-
252","processUptime": 178162}
2021-10-30T05:29:28.158 [Information] Host Status: {"id": "iotblobdatatriggerapp","state": "Running","version":
"3.3.1.0","versionDetails": "3.3.1 Commit hash: bd2b318013cee8d608c88203ec4ee3516d2a9961","platformVersion":
"95.0.7.554","instanceId": "d141dfc91ce38ea7c837f8110af3fb3664d428da4dac25484a7350a5290960d9","computerName": "10-30-5-
252","processUptime": 178162}
```

Home >

IoTBlobDataTriggerApp

Function App

Search (Cmd+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Security
- Events (preview)

Functions

- Functions
- App keys
- App files
- Proxies

Deployment

- Deployment slots
- Deployment Center

Settings

- Configuration

Browse | Refresh | Stop | Restart | Swap | Get publish profile | Reset publish profile | Download app content

Click here to access Application Insights for monitoring and profiling for your app. →

Tags (Edit) : Click here to add tags

Metrics | Features (9) | Notifications (1) | Quickstart