

Indoor real-time position tracking using Azure Maps and IoT Hub

Deployment Instructions

Clone repo

<https://github.com/Azure-Samples/azuremaps-indoor-realtime-position-tracking>

Create resource group in Azure Subscription

To isolate all resources associated with this deployment, create a new resource group in the Azure portal.

Create IoT Hub instance

Create an IoT Hub instance in the Azure portal and make sure to select either the basic tier or higher:

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [IoT Hub](#) >

IoT hub

Microsoft

Basics Networking **Management** Add-ons Tags Review + create

Each IoT hub is provisioned with a certain number of units in a specific tier. The tier and number of units determine the maximum daily quota of messages that you can send. [Learn more](#)

Scale tier and units

Pricing and scale tier * ⓘ

B1: Basic tier ▼

[Learn how to choose the right IoT hub tier for your solution](#)

Number of B1 IoT hub units ⓘ

1

Determines how your IoT hub can scale. You can change this later if your needs increase.

Pricing and scale tier ⓘ	B1	Device-to-cloud-messages ⓘ	Enabled
Messages per day ⓘ	400,000	Message routing ⓘ	Enabled
Cost per month	10.00 USD	Cloud-to-device commands ⓘ	Only available in Free/Standard tiers
Defender for IoT ⓘ	Not eligible	IoT Edge ⓘ	Only available in Free/Standard tiers
Device updates ⓘ	Not eligible	Device management ⓘ	Only

[Review + create](#)

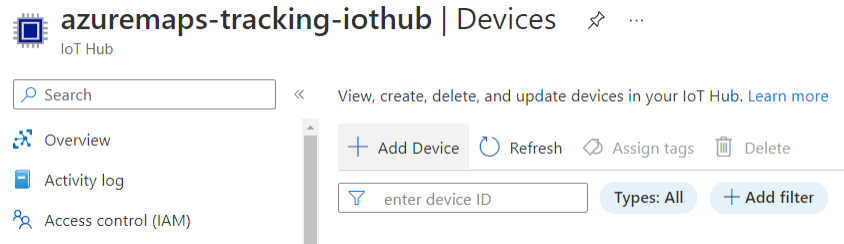
[< Previous: Networking](#)

[Next: Add-ons >](#)

This is important because we are going to be using routing and the free tier only allows 1 route. Plus, the basic tier is also upgradable if needed.

Add Device (e.g., smartphone)

In IoT Hub, select add a new device:



And create your device. You can call it “myiphone” device, for example, since we are going to be using the IoT Plug and Play Application from the Apple (or Google) store.

Create a device

Find Certified for Azure IoT devices in the Device Catalog

Device ID

The ID of the new device

Authentication type

Symmetric key X.509 Self-Signed X.509 CA Signed

Auto-generate keys

☒

Connect this device to an IoT hub

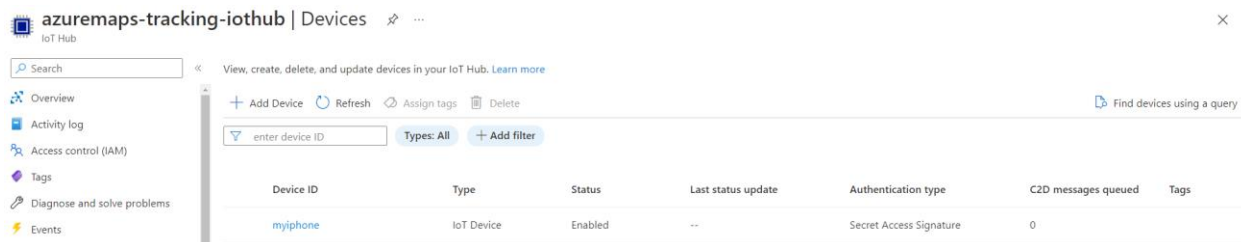
Enable Disable

Parent device

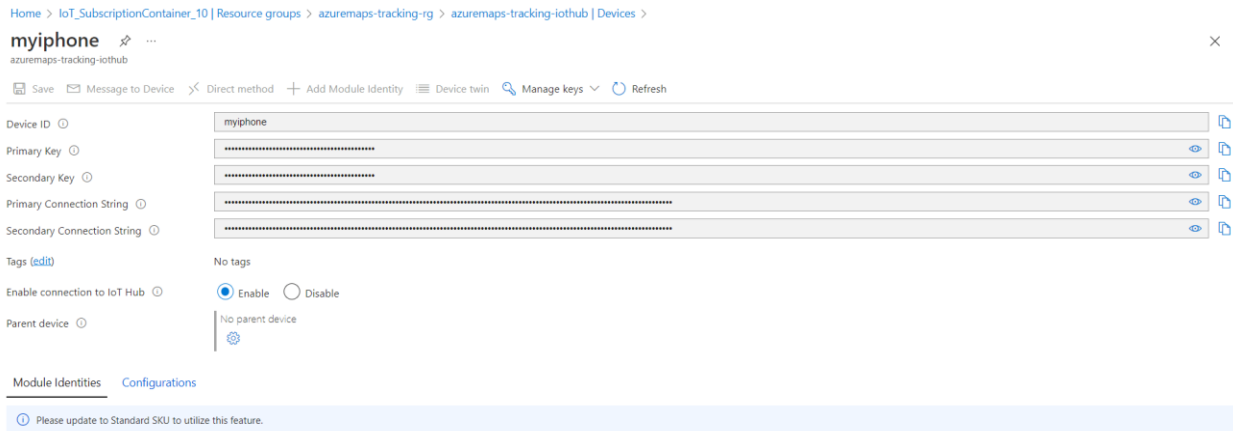
No parent device

[Set a parent device](#)

Once the device is created, click on that device:



and copy the corresponding primary connection string:

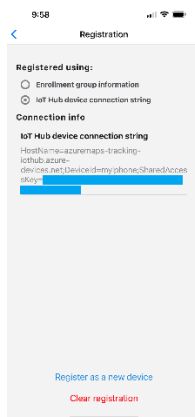


You are going to need this connection string for linking the smart phone application with this device in IoT Hub.

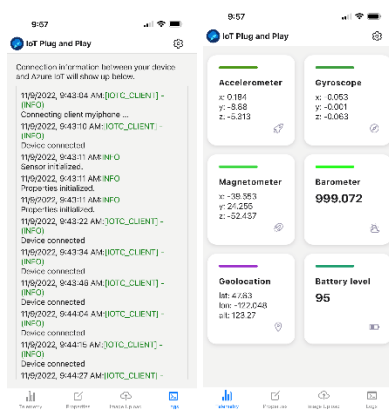
Install IoT Plug and Play Application

On your smartphone, install the “IoT Plug and Play” app. You can find it in the Apple or Google store.

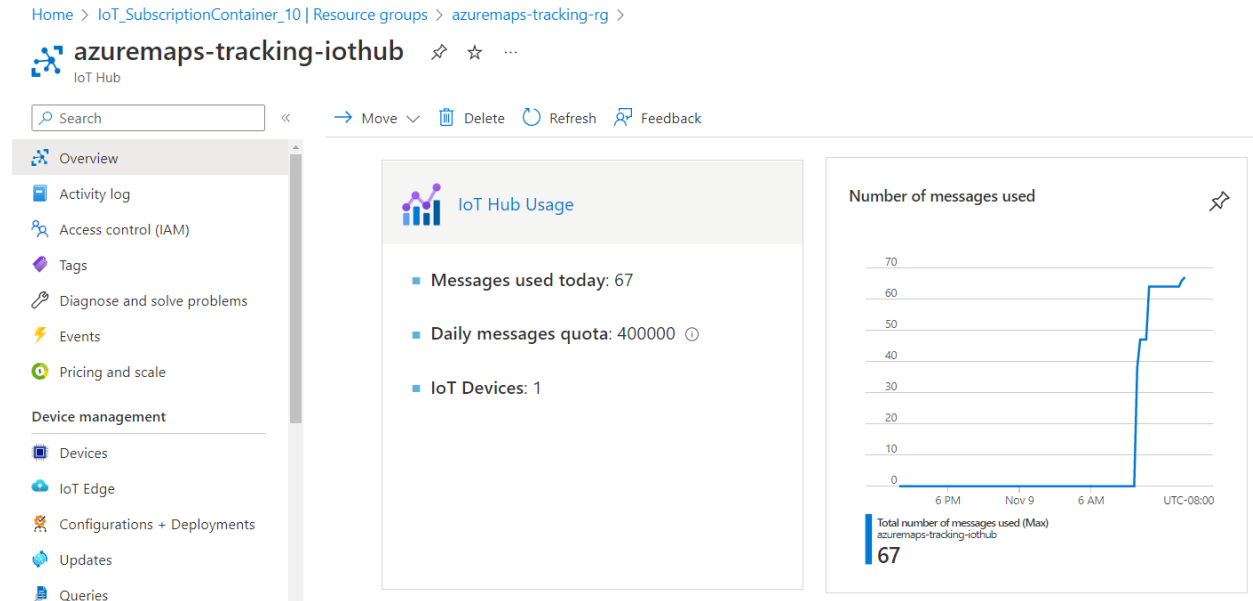
Go to Settings and Registration and paste the connection string from the previous step:



And make sure that you can confirm that it is connected to Azure by checking the “Logs” tab in the application:



You should now be receiving messages from this device in IoT Hub. You can confirm this by checking the “Overview” page and looking at “Number of messages used” and “Connected Devices”.



Next, go to the “Built-in endpoints” under “Hub settings”, scroll down, and copy the “Event Hub-compatible endpoint” connection string. We will use this value later to trigger the Function App whenever new messages arrive to IoT Hub via the built-in Event Hub endpoint.

Home > IoT_SubscriptionContainer_10 | Resource groups > azuremaps-tracking-rg > azuremaps-tracking-iotHub

azuremaps-tracking-iotHub | Built-in endpoints IoT Hub

Search << Save Undo

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events
- Pricing and scale

Device management

- Devices
- IoT Edge
- Configurations + Deployments
- Updates
- Queries

Hub settings

Event Hub-compatible name ⓘ
azuremaps-tracking-iotHub ⓘ

Retain for ⓘ
1 Days

Consumer Groups ⓘ
Consumer Groups
\$Default
Create new consumer group

Event Hub compatible endpoint

Only policies that allow service connect permissions can be selected. These permissions are applied to the built-in endpoint.

Shared access policy ⓘ
iothubowner

Event Hub-compatible endpoint ⓘ
Endpoint=sb://iothub-ns-azuremaps-22532363-dc5580919f.servicebus.windows.net/;SharedAccessKeyName=

Finally, take note of the “Event hub-compatible name” as well.

Create Azure Web PubSub

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

Web PubSub Service ...

Microsoft



Web PubSub Service [Add to Favorites](#)

Microsoft

★ 4.0 (3 Marketplace ratings) | ★ 4.0 (2 external ratings)

Plan

Web PubSub Service


Create

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Web PubSub Service](#) >

+ Web PubSub Service ...

Web PubSub Service

*** Basics** [Networking](#) [Tags](#) [Review + create](#)

Deploy fully managed WebPubSub Service at scale. [Learn more about WebPubSub](#) 

Project Details

Subscription * ⓘ

IoT_SubscriptionContainer_10

Resource group * ⓘ

azuremaps-tracking-rg

[Create new](#)

Service Details

Resource Name *

azuremaps-tracking ✓

.webpubsub.azure.com

Region * ⓘ

East US

Pricing tier * ⓘ

Free

Up to 20 connections, 40,000 KB messages per day included

[Change](#)

[Review + create](#)

[Next : Networking >](#)

[Download a template for automation](#)

Under “Keys”, take a note of the connection string for this service.

Create Storage Account

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

Storage account ...

Microsoft



Storage account [Add to Favorites](#)

Microsoft

★ 4.2 (5322 Marketplace ratings) | ★ 4.2 (3548 external ratings)

Plan

Storage account

Create

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Storage account](#) >

Create a storage account ...

[Basics](#) [Advanced](#) [Networking](#) [Data protection](#) [Encryption](#) [Tags](#) [Review](#)

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *

IoT_SubscriptionContainer_10



Resource group *

azuremaps-tracking-rg

[Create new](#)

Instance details

If you need to create a legacy storage account type, please click [here](#).

Storage account name ⓘ *

azuremapstracking

Region ⓘ *

(US) East US

Performance ⓘ *

☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)

☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ *

Locally-redundant storage (LRS)

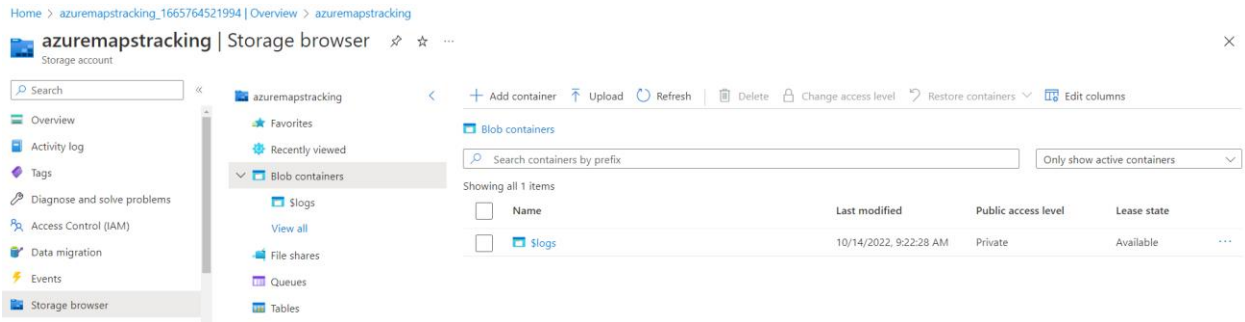
Review

< Previous

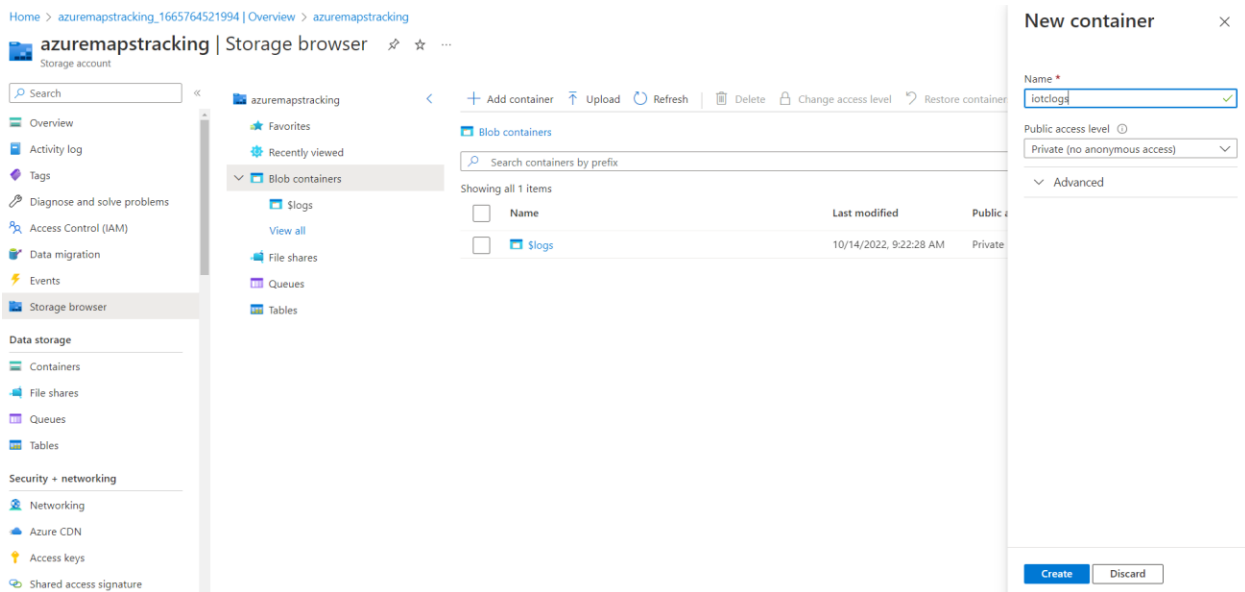
Next : Advanced >

Save the value for the connection string under “Access keys” since you will need it later.

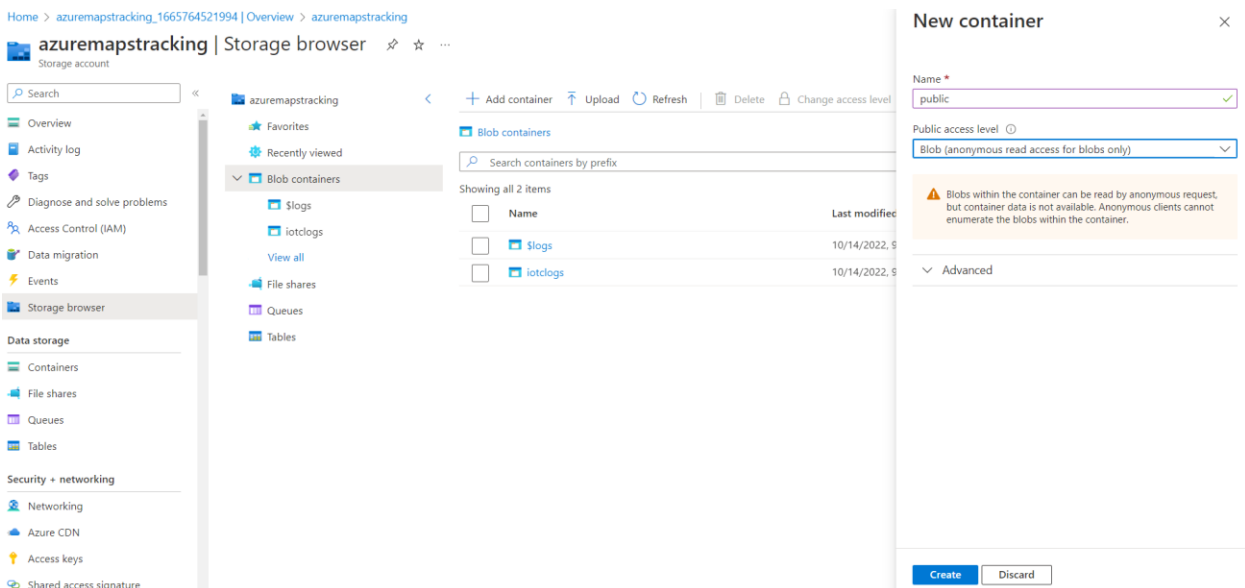
Under “Storage browser”, click on “Blob containers”:



Create a new blob container called “iotclogs” (private):



Create another blob container called “public” with anonymous read access:



Enable CORS:

Home > azuremaptracking_1665764521994 | Overview > azuremaptracking

azuremaptracking | Resource sharing (CORS) ☆ ...

Storage account

Search Save Discard

Data management

- Redundancy
- Data protection
- Object replication
- Blob inventory
- Static website
- Lifecycle management
- Azure search

Settings

- Configuration
- Data Lake Gen2 upgrade
- Resource sharing (CORS)

CORS is an HTTP feature that enables a web application running under one domain to access resources in another domain. Web browsers implement a security restriction known as same-origin policy that prevents a web page from calling APIs in a different domain. CORS provides a secure way to allow one domain (the origin domain) to call APIs in another domain.

You can set CORS rules individually for each of the storage services (i.e. blob, file, queue, table). Once you set the CORS rules for the service, then a properly authenticated request made against the service from a different domain will be evaluated to determine whether it is allowed according to the rules you have specified.

[Learn more about CORS support for Azure Storage](#)

Blob service File service Queue service Table service

Allowed origins	Allowed methods	Allowed headers	Exposed headers	Max age
*	GET			0
	0 selected			0

Install “Azure Storage Explorer”: <https://azure.microsoft.com/en-us/features/storage-explorer/>

Using Azure Storage Explorer, upload the content of the “public” folder in your repo to the corresponding blob container.

Microsoft Azure Storage Explorer

File Edit View Help

EXPLORER

Search for resources

Collapse All Refresh All

- Quick Access
- Emulator & Attached
- Storage Accounts
- Data Lake Storage Gen1 (deprecated)
- IoT_SubscriptionContainer_10 (osnaim@micro)
- Storage Accounts
 - azuremaptracking
 - Blob Containers
 - public
 - File Shares
 - Queues
 - Tables

public

Release Notes: 1.26.0

Upload Download Open Preview New Folder Select All Copy Paste Clone Delete Undelete Manage History Folder Statistics Refresh

Active blobs (default) public

Name	Access Tier	Access Tier Last Modified	Last Modified	Blob Type	Content Type	Size	Status	Remaining Days
Woodland_Park_Zoo					Folder			
azure-maps-layer-legend.min.css	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	text/css	6.07 KiB	Active	
azure-maps-layer-legend.min.js	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	application/javascript	45.09 KiB	Active	
favicon.ico	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	image/x-icon	14.73 KiB	Active	
Zoo_Map.png	Hot (inferred)		10/14/2022 9:44 AM	Block Blob	image/png	2.10 MiB	Active	

Create Azure Maps instance

Home > azuremaps-tracking-rg > Marketplace >

Azure Maps ☆ ...

Microsoft



Azure Maps ❤️ Add to Favorites

Microsoft

Plan

Azure Maps

Create

[Home](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Azure Maps](#) >

Create an Azure Maps Account resource ...

[Basics](#) [Advanced](#) [Identities](#) [Tags](#) [Review + create](#)

Azure Maps is a collection of geospatial services and SDKs that use fresh mapping data to provide geographic context to web and mobile applications. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="IoT_SubscriptionContainer_10"/>
Resource group *	<input type="text" value="azuremaps-tracking-rg"/>
	Create new

Instance details

Name *	<input type="text" value="azuremaps-tracking"/>
Region *	<input type="text" value="East US"/>
Pricing tier *	<input type="text" value="Gen2 (Maps and Location Insights)"/>

[View full pricing details](#)

Terms

Azure Maps shares customer-provided address/location queries ("Queries") with third party TomTom for mapping functionality purposes. Queries are not linked to any customer or end-user when shared with TomTom and cannot be

[Review + create](#)

[< Previous](#)

[Next : Advanced >](#)

Enable CORS:

[Home](#) > [Microsoft.Maps-20221014101719](#) | [Overview](#) > [azuremaps-tracking](#)

azuremaps-tracking | CORS

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Settings

Creator

Authentication

Pricing Tier

Identity

CORS

Shared Access Signature

Save

Discard

Cross-Origin Resource Sharing (CORS) allows JavaScript code running in a browser on an external host to interact with your backend. Specify the origins that should be allowed to make cross-origin calls (for example: [http://example.com:12345](#)). To allow all, use "*" or remove all origins from the list. Slashes are not allowed as part of domain or after TLD.

[Learn more](#)

Allowed origins

*

http://example.com:12345

Save the primary key Azure Maps, which you can find under "Authentication".

Create Azure Function

[Home](#) > [IoT_SubscriptionContainer_10 | Resource groups](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) >

Function App

Microsoft



Function App

[Add to Favorites](#)

Microsoft

★ 4.1 (4251 Marketplace ratings) | ★ 4.1 (2834 external ratings)

Plan

Function App

Create

[Home](#) > [azuremaps-tracking](#) > [azuremaps-tracking-rg](#) > [Marketplace](#) > [Function App](#) >

Create Function App

Project Details

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

IoT_SubscriptionContainer_10



Resource Group * ⓘ

azuremaps-tracking-rg

[Create new](#)

Instance Details

Function App name *

azuremaps-tracking

.azurewebsites.net

Publish *



Code



Docker Container

Runtime stack *

Node.js

Version *

16 LTS

Region *

East US

Operating system

The Operating System has been recommended for you based on your selection of runtime stack.

Operating System *



Linux



Windows

[Review + create](#)

[< Previous](#)

[Next : Hosting >](#)

And make sure you use the same storage account we created previously:

Home > azuremaps-tracking-rg > Marketplace > Function App >

Create Function App ...

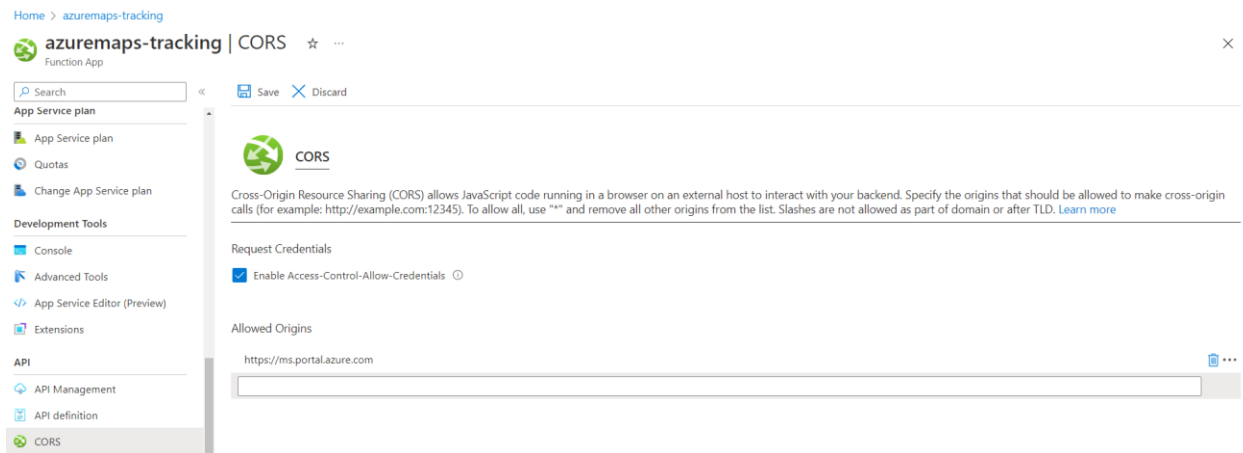
Basics Hosting Networking Monitoring Deployment Tags Review + create

Storage

When creating a function app, you must create or link to a general-purpose Azure Storage account that supports Blobs, Queue, and Table storage.

Storage account *
 [Create new](#)

Once the Function App is created, enable CORS:



Then, under “Configuration”, add a new entry for “WebPubSubConnectionString” with the corresponding value that we saved previously.

Also, add another entry for “EventHubConnectionString” and use the value we saved when we created the IoT Hub instance.

Application settings Function runtime settings General settings

Application settings

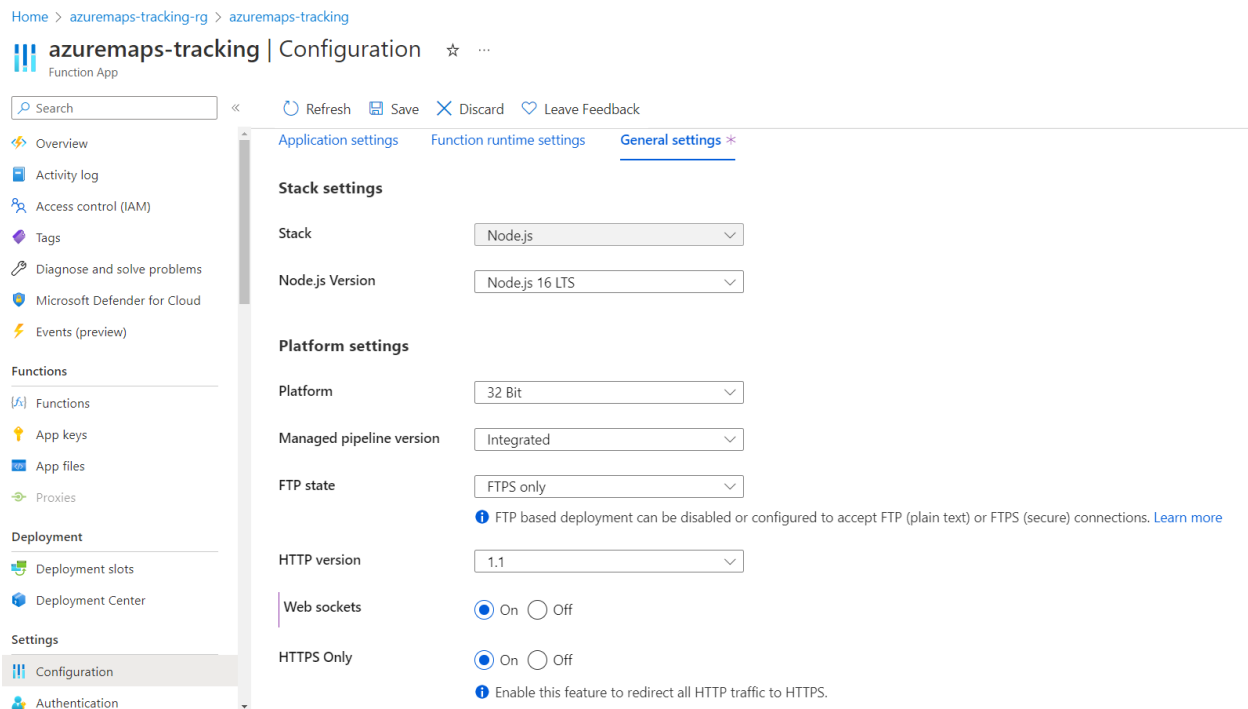
Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to display them in plain text in your browser by using the controls below. Application Settings are exposed as environment variables for access by your application at runtime. [Learn more](#)

+ New application setting Show values Advanced edit

Filter application settings

Name	Value	Source	Deployment slot setting	Delete	Edit
APPINSIGHTS_INSTRUMENTATIONKEY	Hidden value. Click to show value	App Service			
APPLICATIONINSIGHTS_CONNECTION_STRING	Hidden value. Click to show value	App Service			
AzureWebJobsStorage	Hidden value. Click to show value	App Service			
EventHubConnectionString	Endpoint=sb://iothub-ns-azuremaps-221	App Service			
FUNCTIONS_EXTENSION_VERSION	Hidden value. Click to show value	App Service			
FUNCTIONS_WORKER_RUNTIME	Hidden value. Click to show value	App Service			
WebPubSubConnectionString	Hidden value. Click to show value	App Service			
WEBSITE_CONTENTAZUREFILECONNECTIONSTI	Hidden value. Click to show value	App Service			
WEBSITE_CONTENTSHARE	Hidden value. Click to show value	App Service			

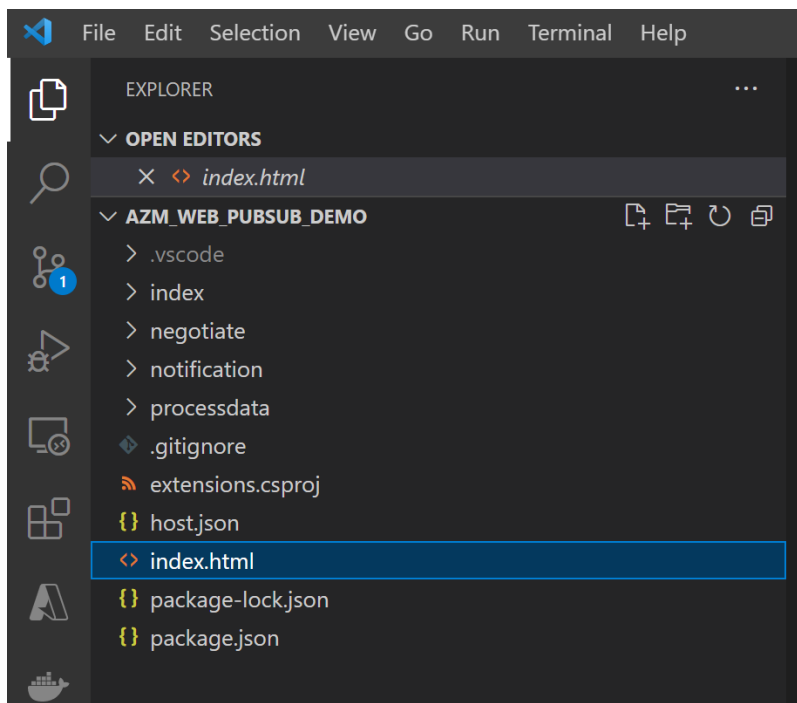
Also, under “Configuration”, enable Web sockets:



Modify Function App

We will now modify the Azure Function by deploying the code we cloned from the repo.

Open the following folder using VS Code (**AZM_WEB_PUBSUB_DEMO**, which you can find under **realtime-azuremaps-update-iothubdemo\AzM_Web_PubSub_Demo-v02**):



Let's start with "index.html" in the main folder.

First, replace "<YOUR-BLOB-STORAGE-URL>" with the corresponding value, which you can get from Settings/Endpoints/Blob service in the Azure portal. There should be 6 instances of it that you need to replace in this file.

For example:

```
<html>
  <head>
    <title>Azure Maps Web PubSub</title>
    <link rel="shortcut icon" href="https://azuremapstracking.blob.core.windows.net/public/favicon.ico"/>
    <meta charset="utf-8">
```

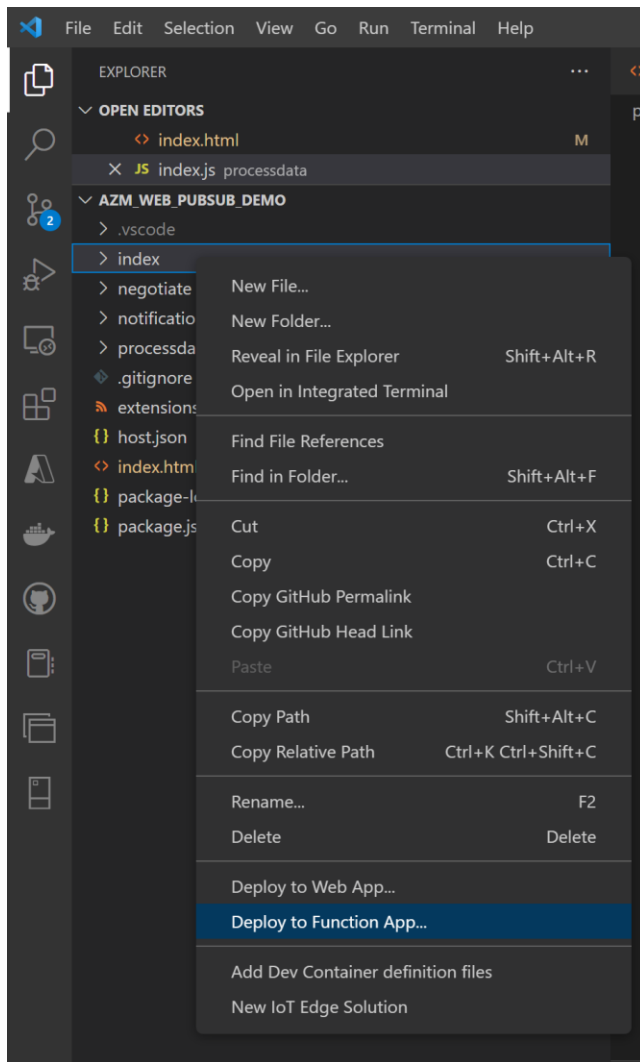
Then, replace "<YOUR-AZURE-MAPS-KEY>" with the map key you had saved previously.

Finally, edit the **function.json** file under the "notification" folder and add the corresponding Event Hub name that we saved when we created the IoT Hub instance. For example:

```
src > realtime-azuremaps-update-iotdemo > AzM_Web_PubSub_Demo-v02 > AzM_Web_PubSub_Demo > notification > {} function.json >
1  {
2    "bindings": [
3      {
4        "type": "eventHubTrigger",
5        "name": "myEventHubMessage",
6        "direction": "in",
7        "eventHubName": "azuremaps-tracking-iotdemo",
8        "connection": "EventHubConnectionString",
9        "cardinality": "many",
10       "consumerGroup": "$Default",
11       "dataType": "string"
12     },
13     {
14       "type": "webPubSub",
15       "name": "actions",
16       "hub": "notification",
17       "direction": "out"
18     }
19   ]
20 }
```

Deploy Function App to Azure

We now need to deploy these functions to the Azure Function we created previously. We will do this by right clicking on the corresponding folder and selecting deploy to Function App:



This step will deploy 4 functions to the Function App: **index**, **negotiate**, **notification**, and **processdata**.

```
TERMINAL  AZURE  JUPYTER  DEBUG CONSOLE  PROBLEMS  OUTPUT  Azure Functions
11:11:57 AM azuremaps-tracking: Added app setting WEBSITE_RUN_FROM_PACKAGE to improve performance of function app. Learn more
here: https://aka.ms/AA8vxc0
11:12:01 AM azuremaps-tracking: Starting deployment...
11:12:08 AM azuremaps-tracking: Creating zip package...
11:12:17 AM azuremaps-tracking: Zip package size: 2.9 MB
11:12:22 AM azuremaps-tracking: Updating submodules.
11:12:25 AM azuremaps-tracking: Preparing deployment for commit id 'a6e3f01268'.
11:12:29 AM azuremaps-tracking: Skipping build. Project type: Run-From-Zip
11:12:30 AM azuremaps-tracking: Skipping post build. Project type: Run-From-Zip
11:12:30 AM azuremaps-tracking: Triggering recycle (preview mode disabled).
11:12:31 AM azuremaps-tracking: Deployment successful.
11:13:16 AM azuremaps-tracking: Started postDeployTask "npm install (functions)".
11:13:27 AM azuremaps-tracking: Syncing triggers...
11:13:37 AM azuremaps-tracking: Querying triggers...
11:13:40 AM azuremaps-tracking: HTTP Trigger Urls:
index: https://azuremaps-tracking.azurewebsites.net/api/index
negotiate: https://azuremaps-tracking.azurewebsites.net/api/negotiate
notification: https://azuremaps-tracking.azurewebsites.net/api/notification

[Azurite Table Service] [Azurite Queue Service] [Azurite Blob Service] Ln 125, Col 70 Spaces: 4 UTF-8 CRLF HTML
```

Home > azuremaps-tracking-rg > azuremaps-tracking

{fx} azuremaps-tracking | Functions ...
Function App

Search << + Create Refresh Delete

⚠ Your app is currently in read only mode because you are running from a package file. To make any changes update the content in your zip file and WEBSITE_RUN_FROM_PACKAGE app setting.

Filter by name...

<input type="checkbox"/> Name ↑↓	Trigger ↑↓	Status ↑↓
<input type="checkbox"/> index	HTTP	Enabled
<input type="checkbox"/> negotiate	HTTP	Enabled
<input type="checkbox"/> notification	HTTP	Enabled
<input type="checkbox"/> processdata	Timer	Enabled

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Microsoft Defender for Cloud
Events (preview)
Functions
[fx] Functions
App keys
App files

Update IoT Hub message routing endpoints

Now, let's go back to our IoT Hub instance and configure message routing so that messages can flow to the corresponding consumers.

First, create a new endpoint called "iotclogs" that is going to be pointing to the blob container that we had created previously:


Routes Custom endpoints

Choose which Azure services will receive messages

+ Add Synchronize keys

- Event hubs
- Service bus queue
- Service bus topic
- Storage

Pick a container by navigating to your storage account:

 **Add a storage endpoint** ...

Route your telemetry and device messages to Azure Storage.

Endpoint name * ⓘ
endpoint name

Azure Storage account and container

Create a new container, or choose an existing one that shares a subscription with this IoT hub.

Azure Storage container

Pick a container

and selecting "iotclogs":

Containers ...

azuremapstracking

+ Container Refresh

Search containers by prefix

Name

\$logs

azure-webjobs-eventhub

azure-webjobs-hosts

azure-webjobs-secrets

iotclogs

public

Repeat this process if you would like to create another route for pushing messages to an Event Hub (that we could then use to connect to Azure Data Explorer if so desired).

Make sure that there is a built-in route enabled for events, so that telemetry can keep flowing to the Function App listening to the built-in Event Hub.

azuremaps-tracking-iothub | Message routing ✕ ...

Send data from your devices to endpoints that you choose.

Routes Custom endpoints Enrich messages

Create an endpoint, and then add a route (you can add up to 100 routes from each IoT hub). Since routing is based on a matching query, a message can be sent to multiple endpoints. Messages that don't match a query are automatically sent to messages/events if you've enabled the fallback route. When you create new endpoints and routes, messages stop flowing to the built-in endpoint unless you create a separate route and direct them there. If no routes to the built-in endpoint exist, enabling a fallback route will direct any messages that don't match a route query to that endpoint. [Learn more](#)

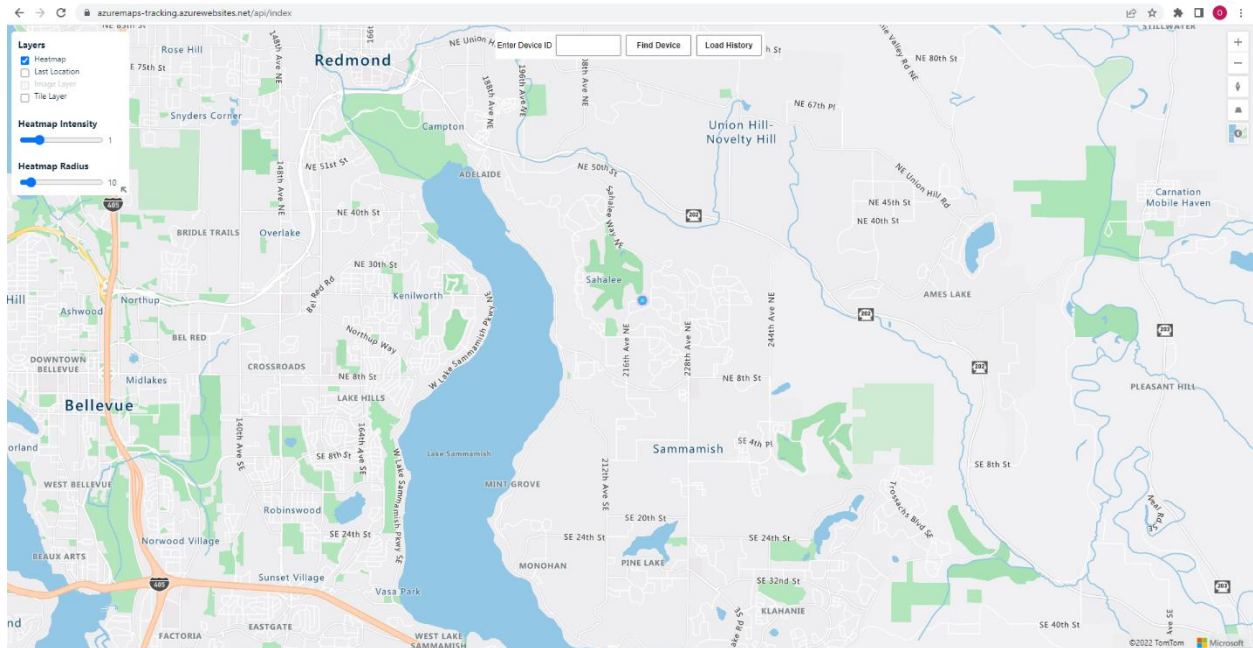
disable fallback route

+ Add Test all routes Delete

<input type="checkbox"/> Name	Data Source	Routing Query	Endpoint	Enabled
<input type="checkbox"/> blob-iotclogs-route	DeviceMessages	true	iotclogs	true
<input type="checkbox"/> built-in-route	DeviceMessages	true	events	true
<input type="checkbox"/> eventhub-route	DeviceMessages	true	eventhub-route	true

View data in your map

You should now be ready to start seeing data points in your map! Just make sure that the IoT Plug and Play application is open in your smartphone.



Just go to the URL that corresponds to the “index” function that you deployed to the Function App:

[Home](#) > [azuremaps-tracking-rg](#) > [azuremaps-tracking | Functions](#) >

index
Function

Enable Disable Delete Get Function URL Refresh

Overview

Developer

Code + Test

Integration

Monitor

Function Keys

Get Function Url

Copied

OK

Resource group (move) : [azuremaps-tracking-rg](#)

Subscription (move) : [IoT SubscriptionContainer 10](#)

Subscription ID : e195708a-04a1-404f-bcdc-835381f2576f

Congratulations! You have completed this tutorial 🎉