Contents

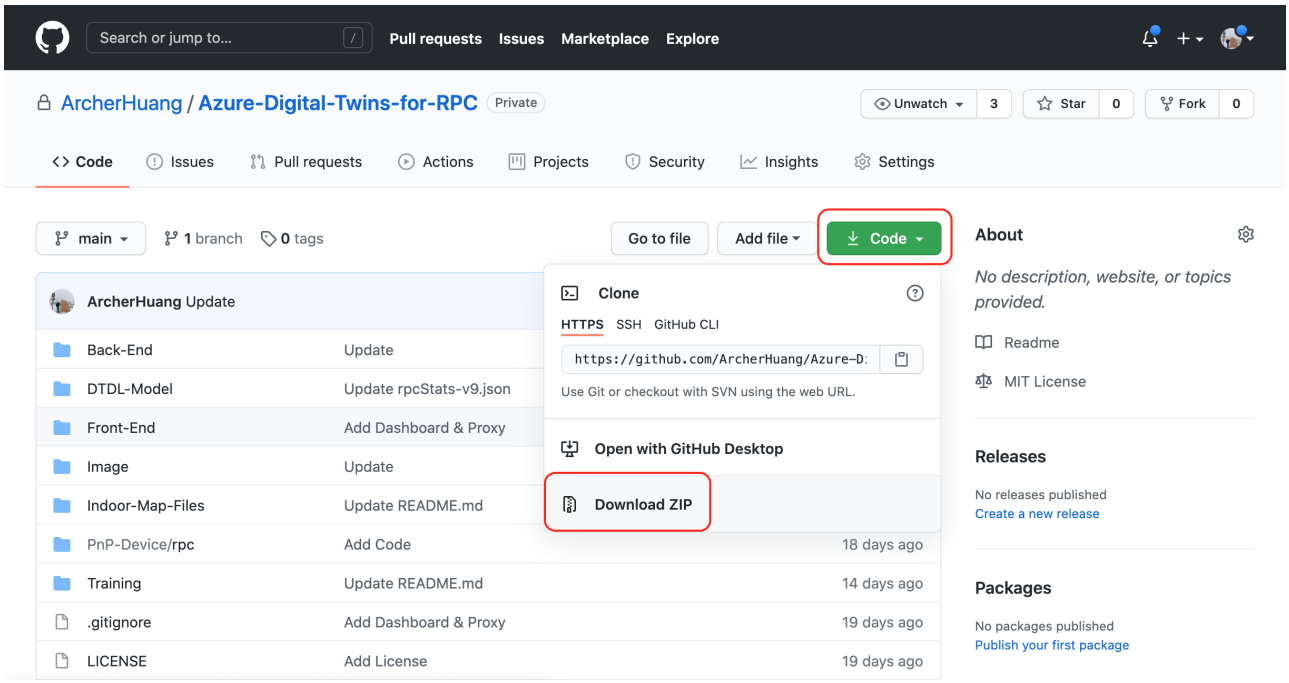
Prerequisites

- Cloud
    - Azure account
- IDE
    - Azure Funciton
        - Visual Studio Community 2019
    - Dashboard & Proxy
        - Visual Studio Code
- Package and Language Version
    - Azure Funciton
        - Azure.Core v1.6.0
        - Azure.DigitalTwins.Core v1.2.0
        - Azure.Identity v1.3.0
        - Microsoft.Net.Sdk.Funtions v3.0.7
        - System.Net.Http v4.3.4
    - Dashboard & Proxy
        - npm
        - Node.js v10.16.0
        - vue v2.6.12
- CLI
    - Azure CLI
- API Client
    - Postman

SOP for Initiating Services

1. Download Sample Code

- 1.1 Download project
  - https://github.com/ArcherHuang/Azure-Digital-Twins-for-RPC
- 1.2 Click `Code` > `Download ZIP`



- 1.3 Unzip `Azure-Digital-Twins-for-RPC-main.zip`

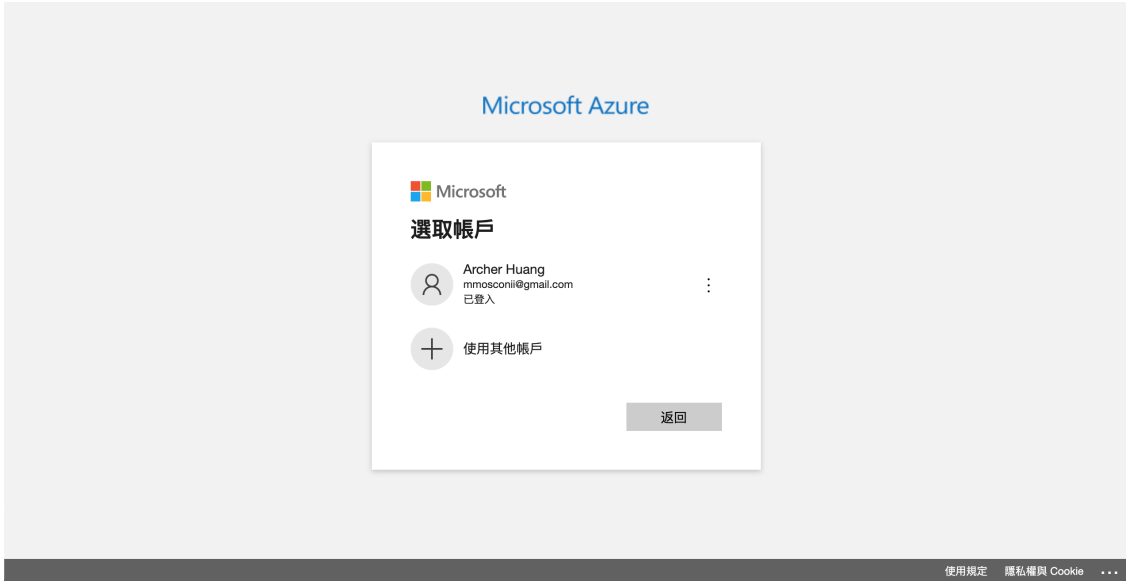
2. Install the Azure CLI and Login

- 2.1 Install the Azure CLI
  - Reference documentation
- 2.2 Sign in with Azure CLI
  - Run the login command

```
az login
```

  - If the CLI can open your default browser, it will do so and load an Azure sign-in page.

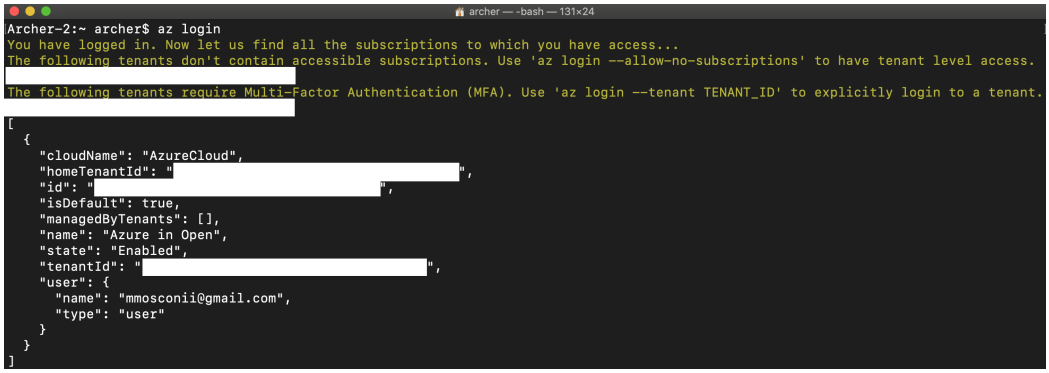
- Sign in with your account credentials in the browser.



- Login Success
  - Browser

> **You have logged into Microsoft Azure!**
>
> You can close this window, or we will redirect you to the Azure CLI documents in 10 seconds.

  - Terminal



3. Create a resource group

- Azure CLI

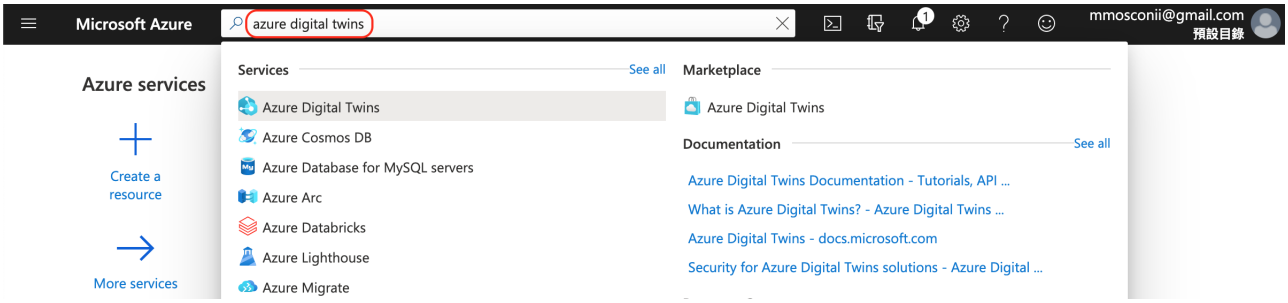
```
az group create --name <ResourceGroup> --location <Region>
```

- e.g.

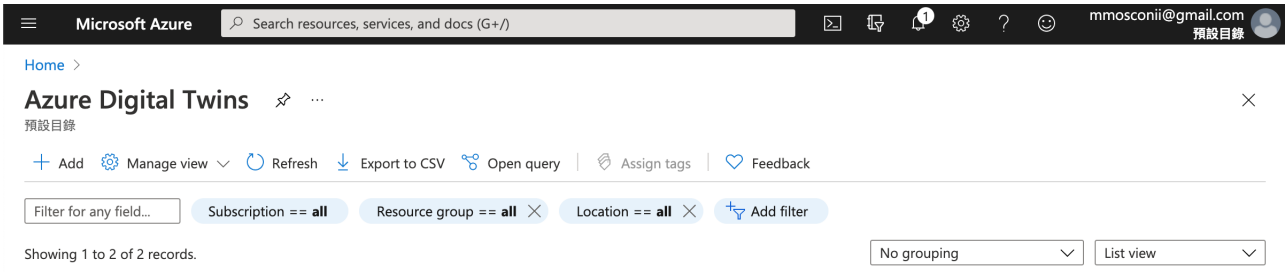
```
az group create --name rpc-adt-rg --location eastus
```

4. Create Azure Digital Twin

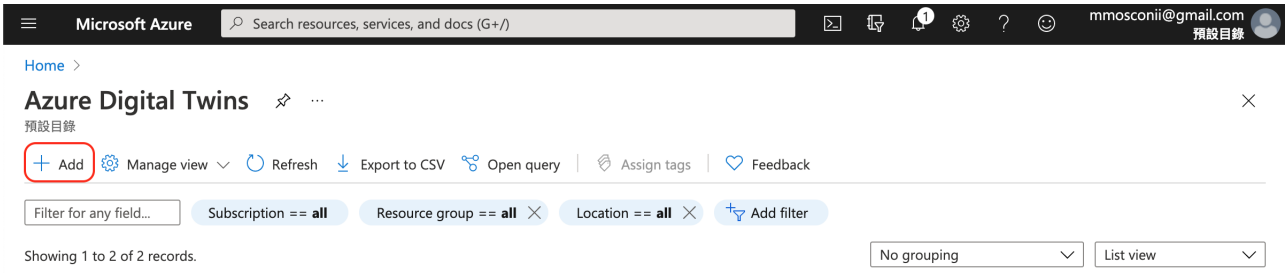
- 4.1 Search `Azure Digital Twins`



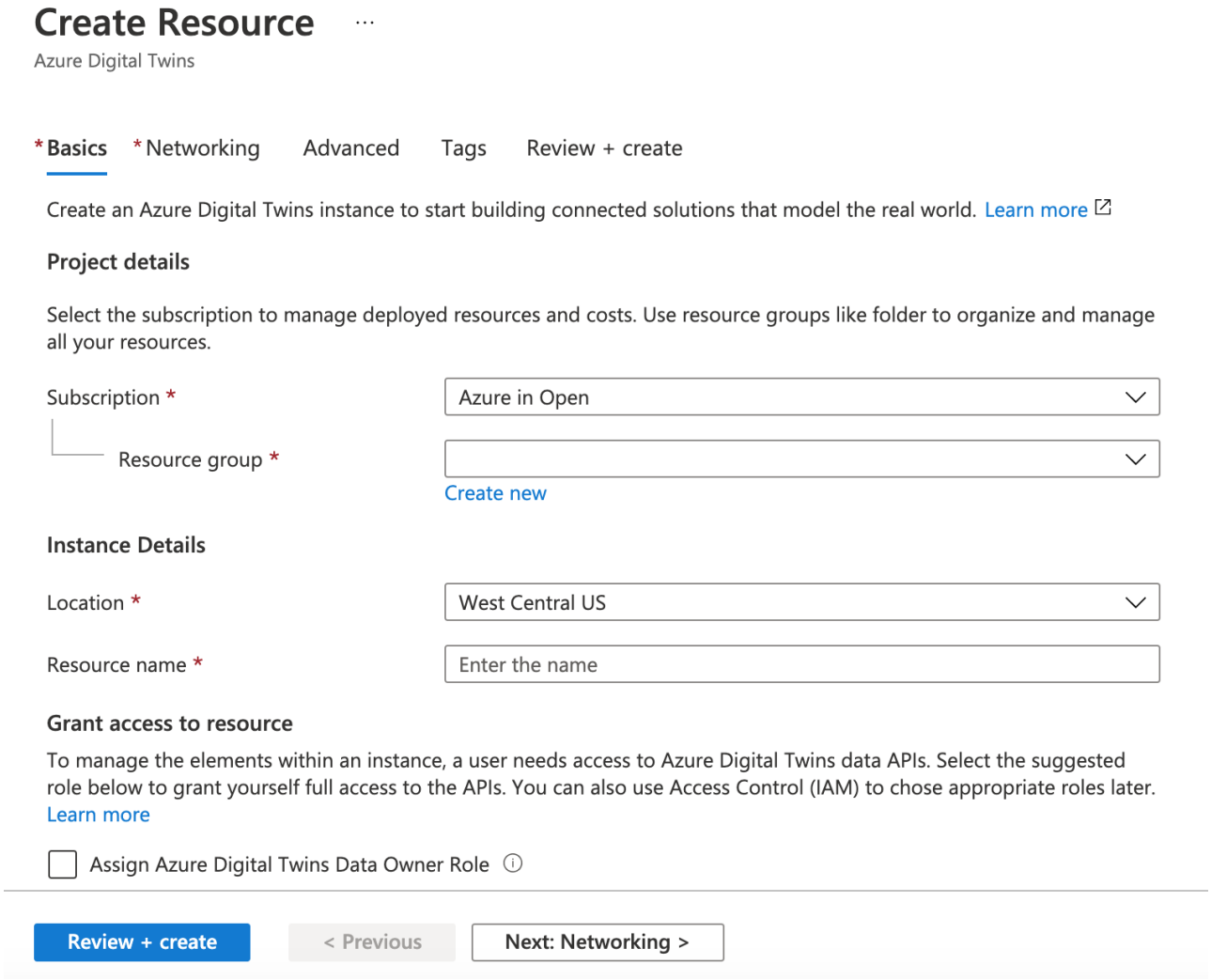
- 4.2 Select `Azure Digital Twins`



- 4.3 Click `Add`



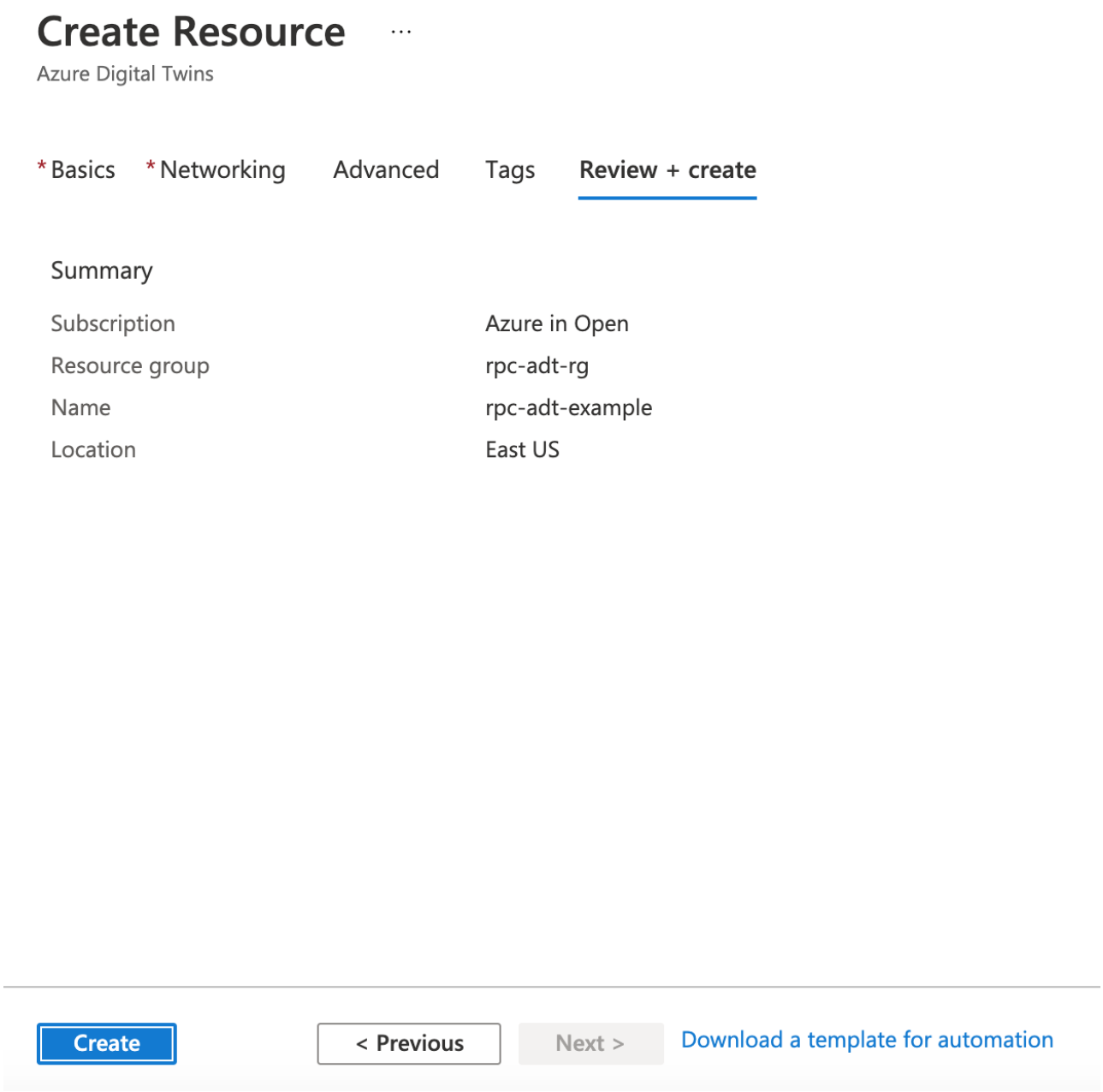
- 4.4 Input data



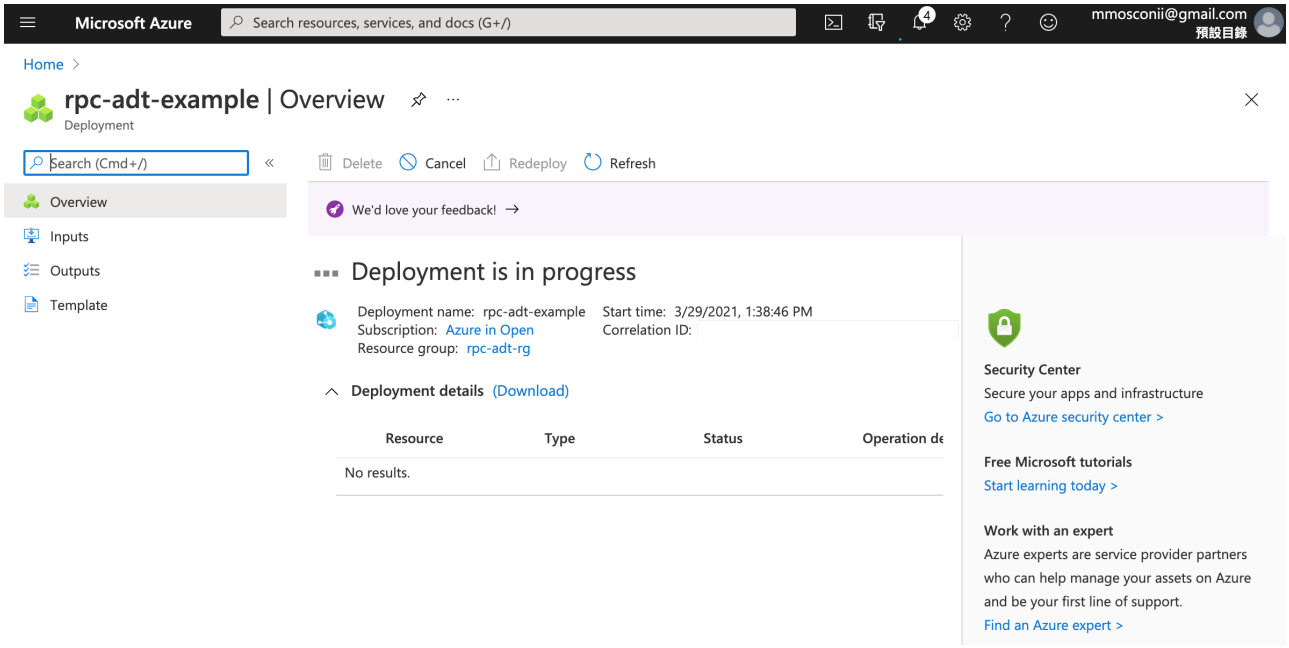
  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name, this example uses `rpc-adt-rg`
  - `Location` field
    - This example uses `East US`
  - `Resource name` field
    - Please use a recognizable name, this example uses `rpc-adt-example`

- 4.5 When the input is complete, please click the `Review + create` button

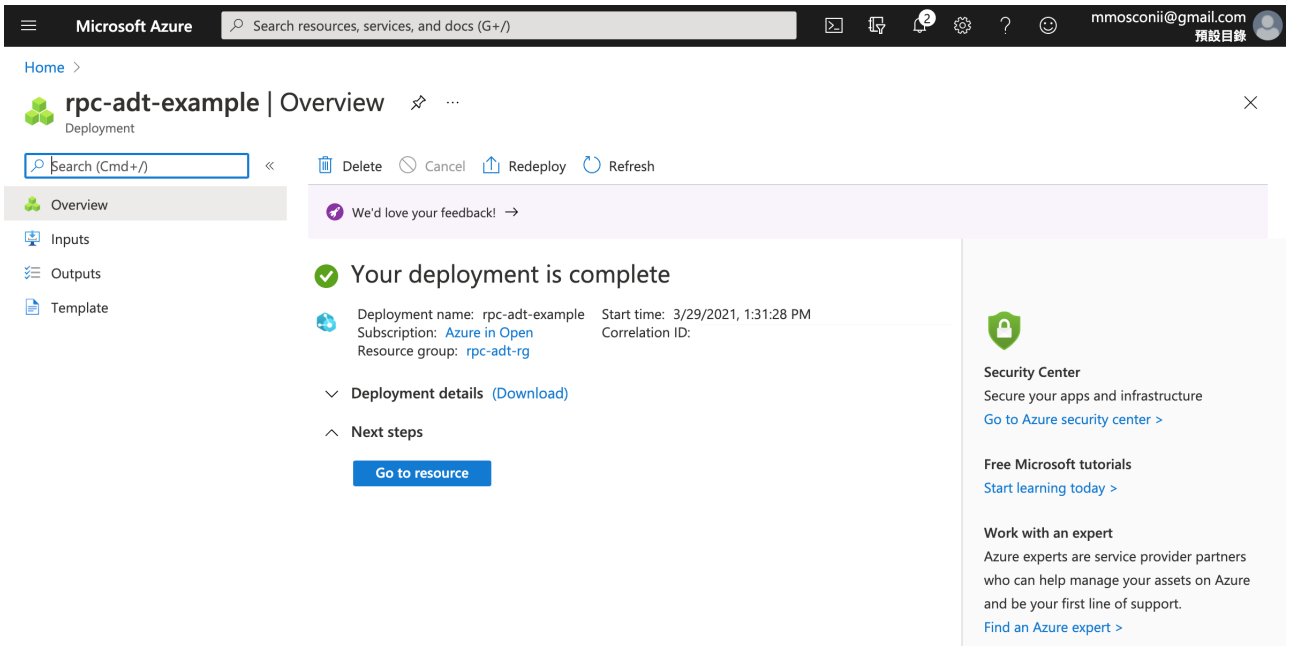
- 4.6 Review your settings and select `create`



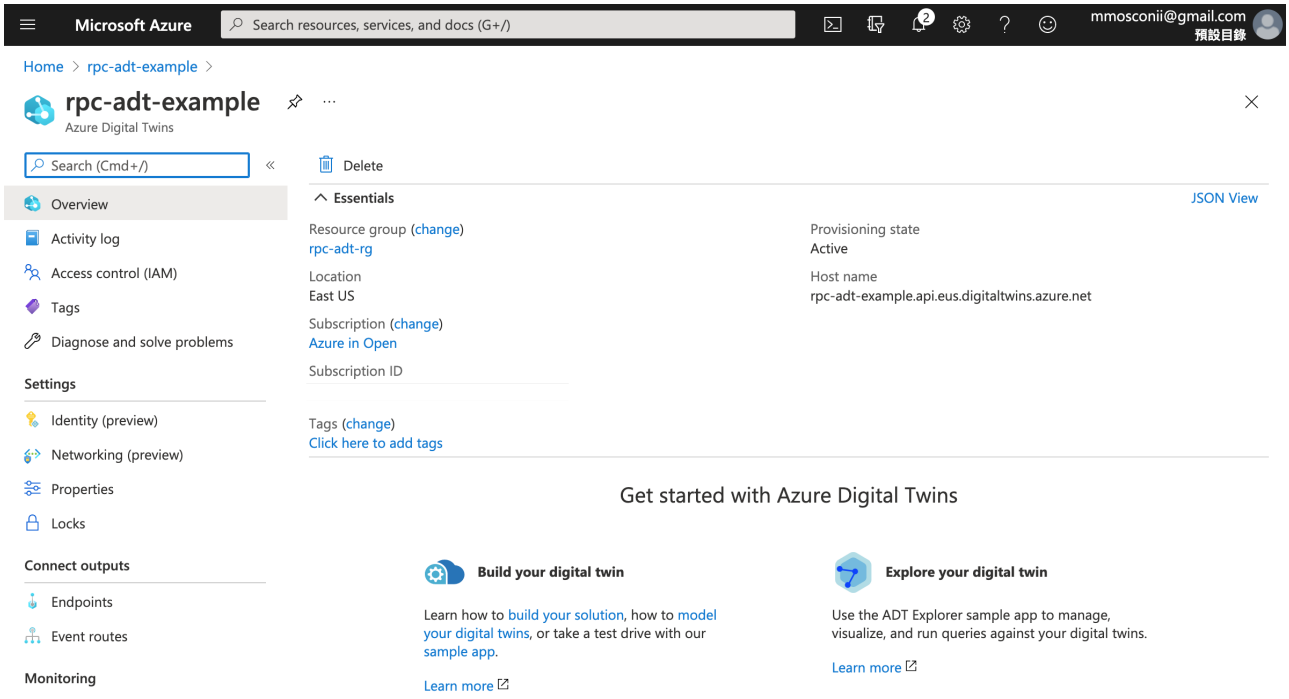
- 4.7 Wait for the request to process



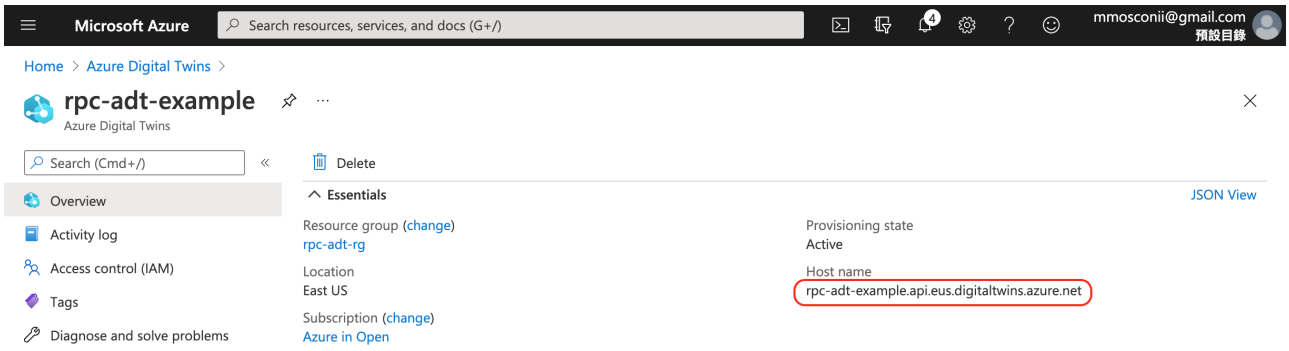
- 4.8 Operation completes



- 4.9 Once deployment complete click on `Go to resource` button

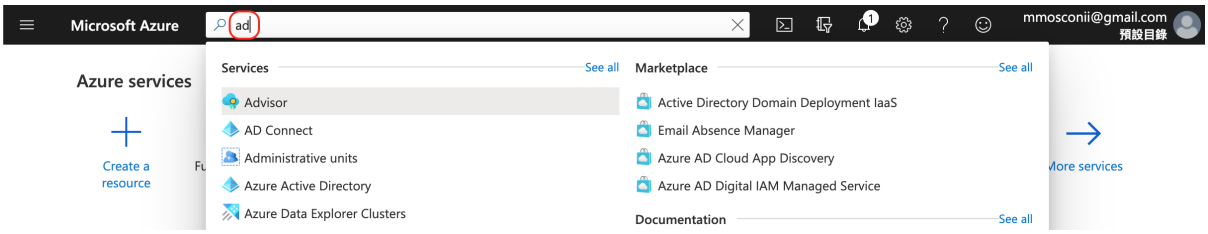


- 4.10 Get `Azure Digital Twins Service URL`

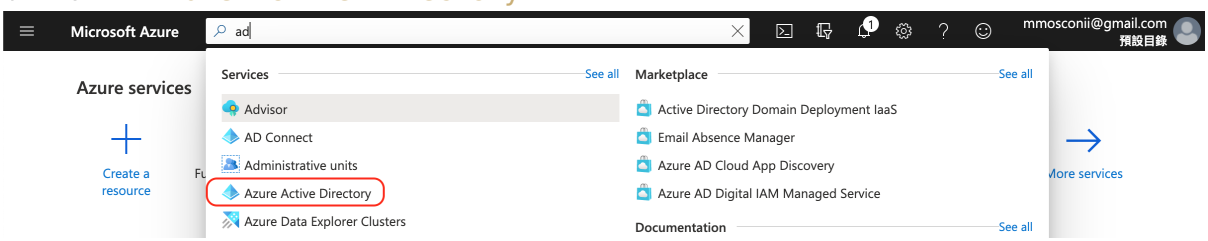


5. Set AD for Azure Map

- 5.1 Open `Azure Active Directory`
  - 5.1.1 Search `ad`

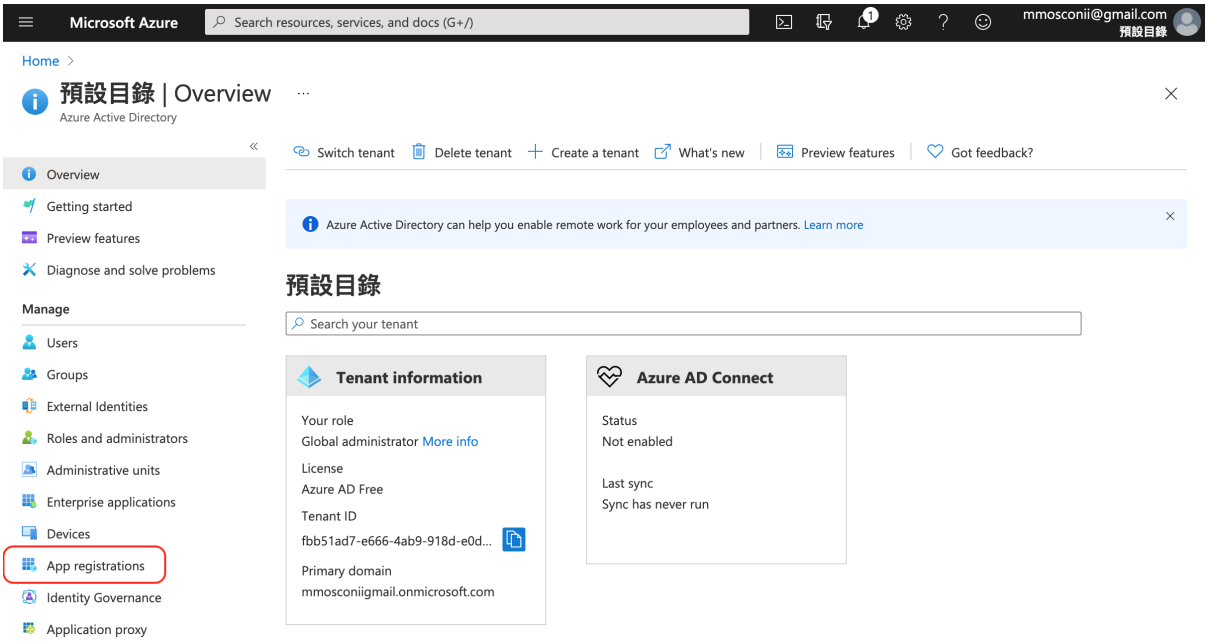


  - 5.1.2 Select `Azure Active Directory`

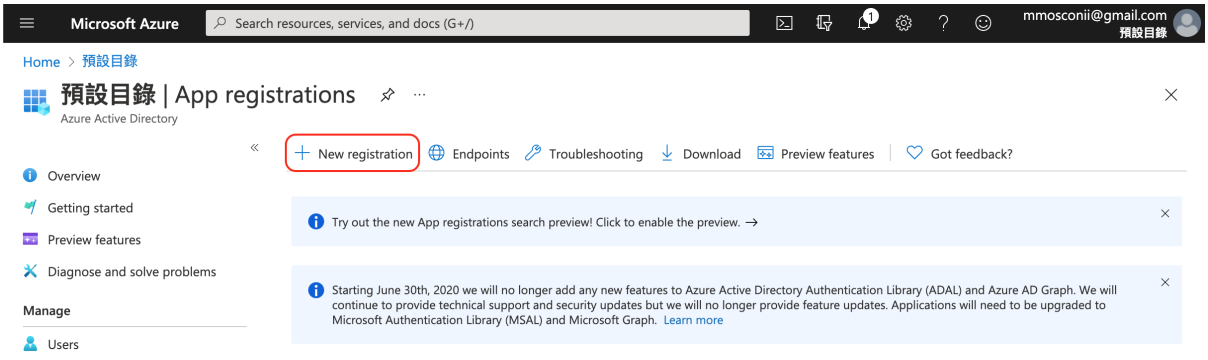


- 5.2 Create App registrations

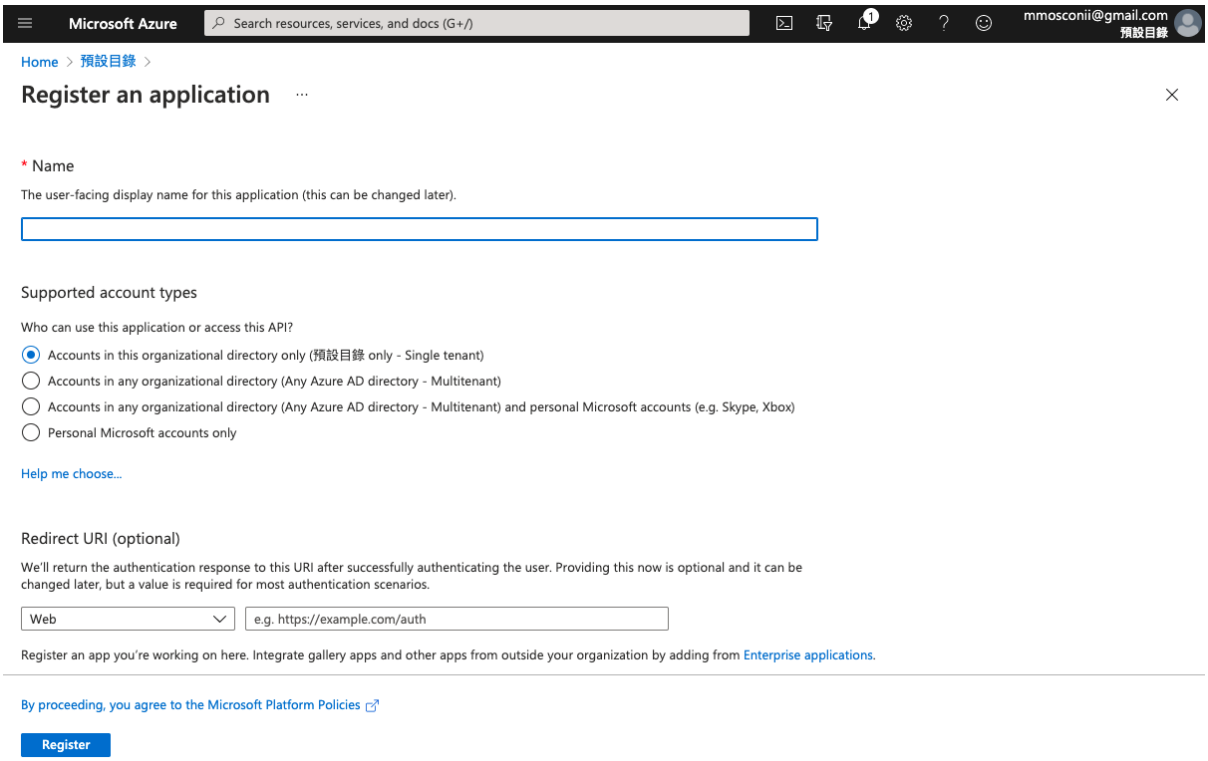
- 5.2.1 Click `App registrations`



- 5.2.2 Click `+ New registration`

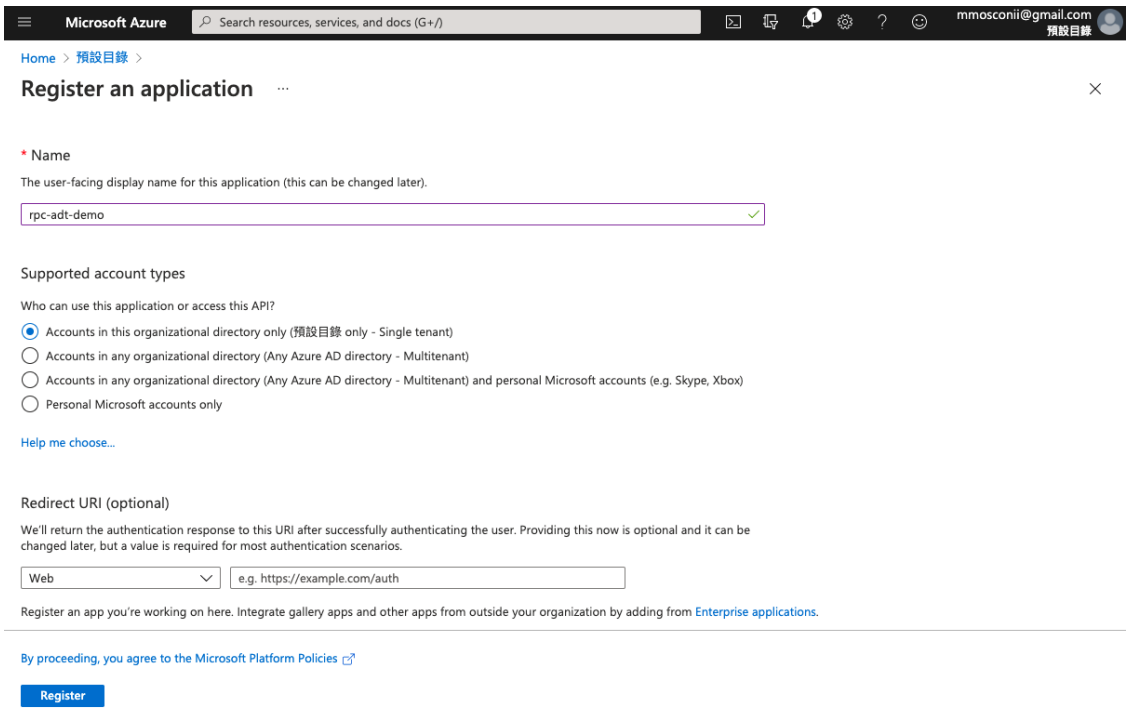


- 5.2.3 Input data

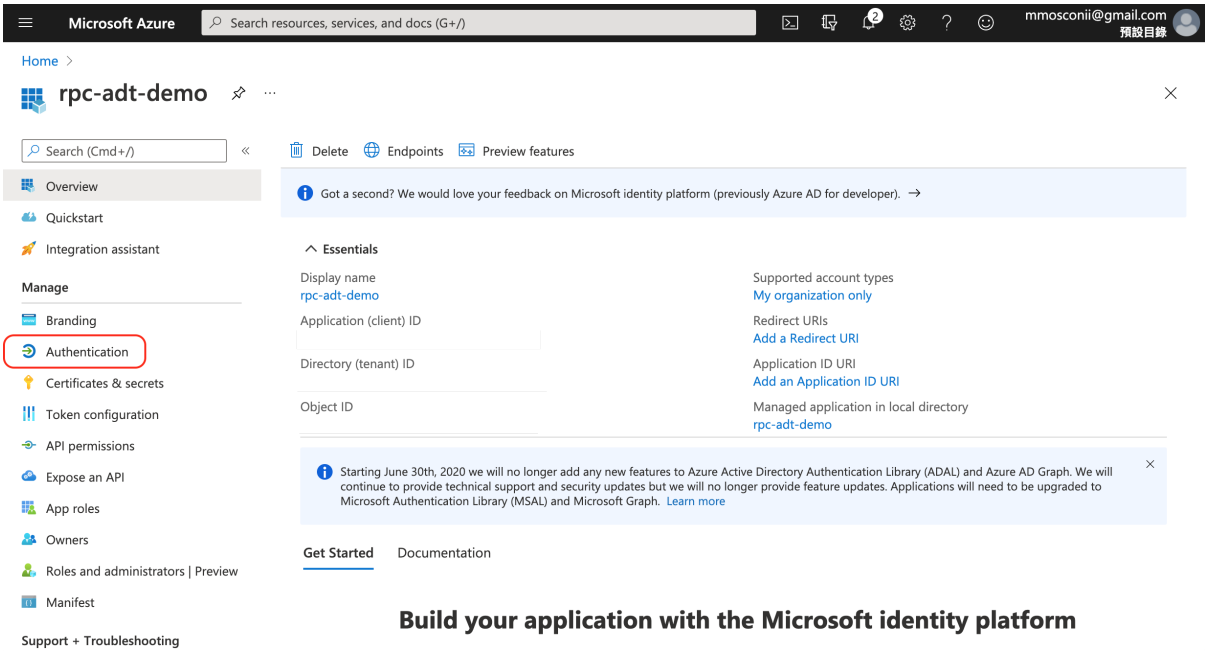


  - `Name` field
    - Please use a recognizable name, this example uses `rpc-adt-demo`

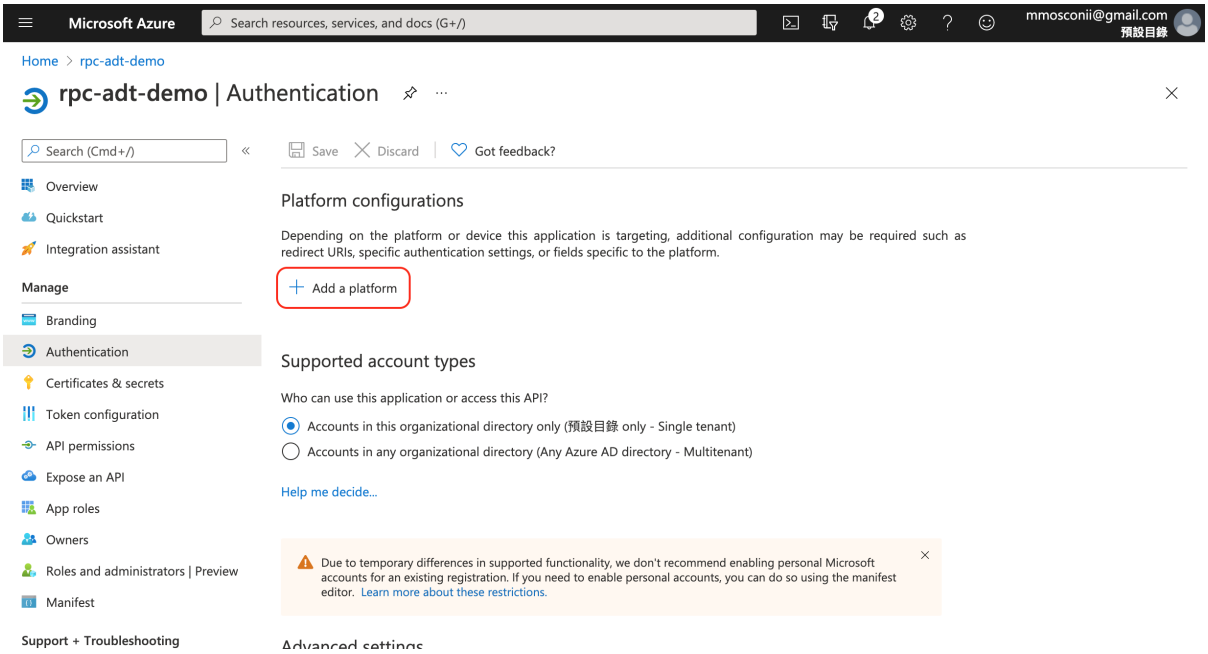
- `Supported account types` field
  - Select `Accounts in this organizational directory only (預設目錄 only – Single tenant)`
- When the input is complete, please click the `Register` button



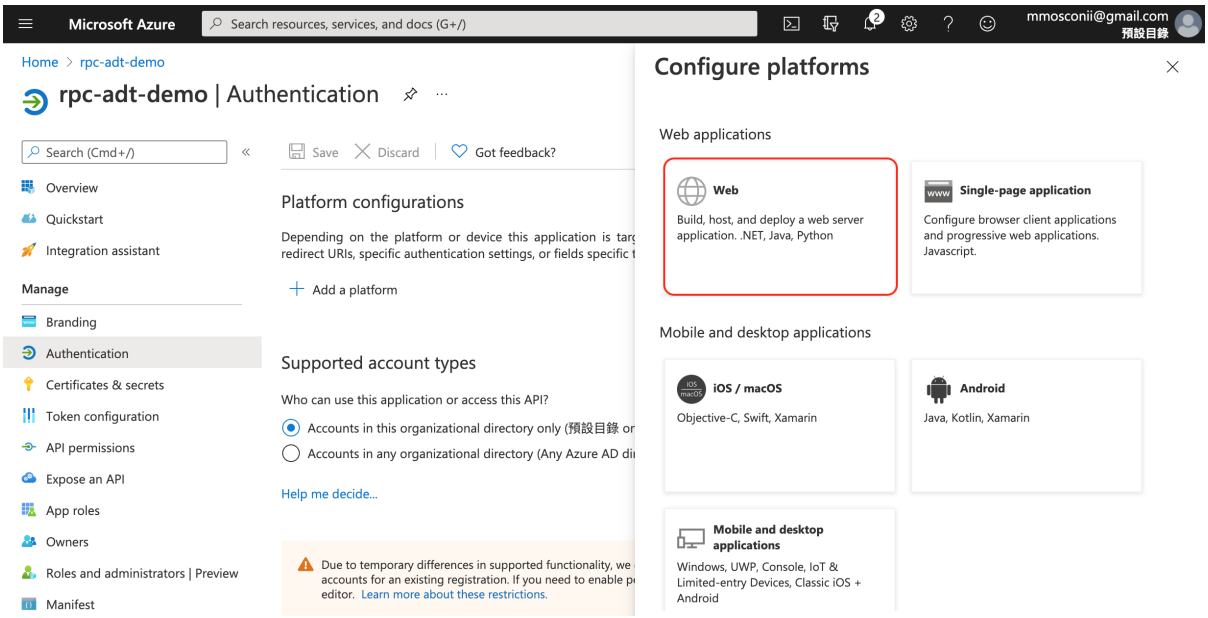
- 5.3 Authentication
  - 5.3.1 Click `Authentication`



- 5.3.2 Click `+ Add a platform`

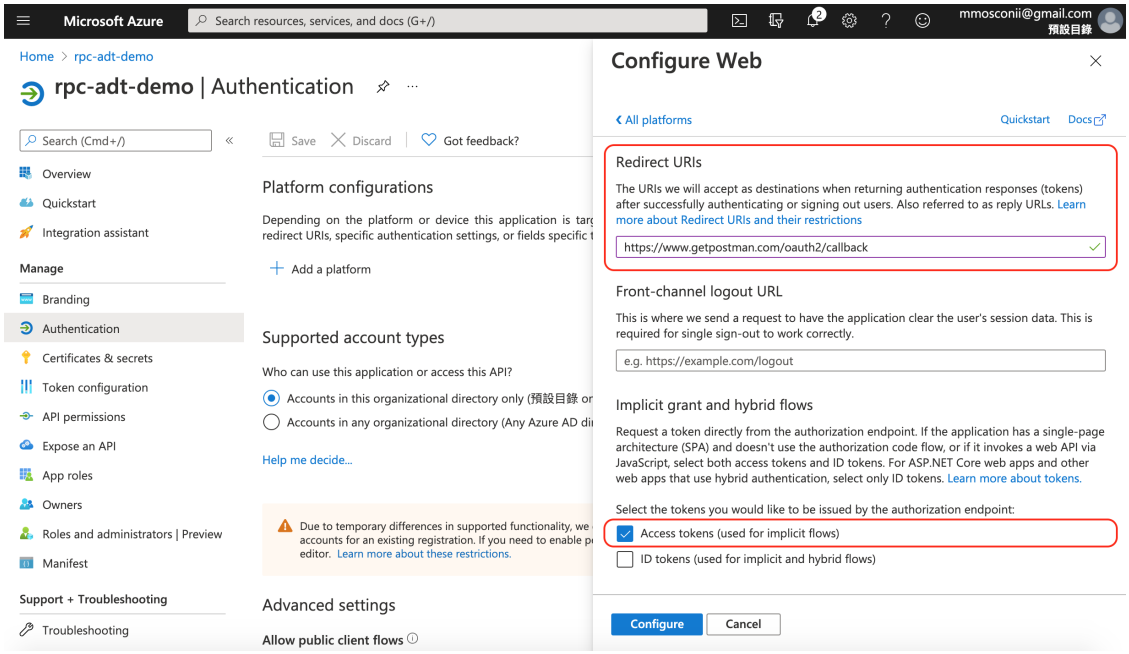


- 5.3.3 Click `Web`

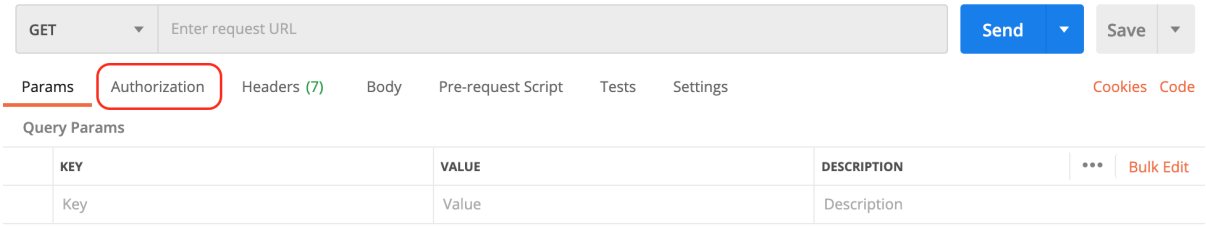


- 5.3.4 Configure Web
  - `Redirect URIs` field
    - Please input `https://www.getpostman.com/oauth2/callback`
  - `Implicit grant and hybrid flows`
    - Select `Access tokens (used for implicit flows)`

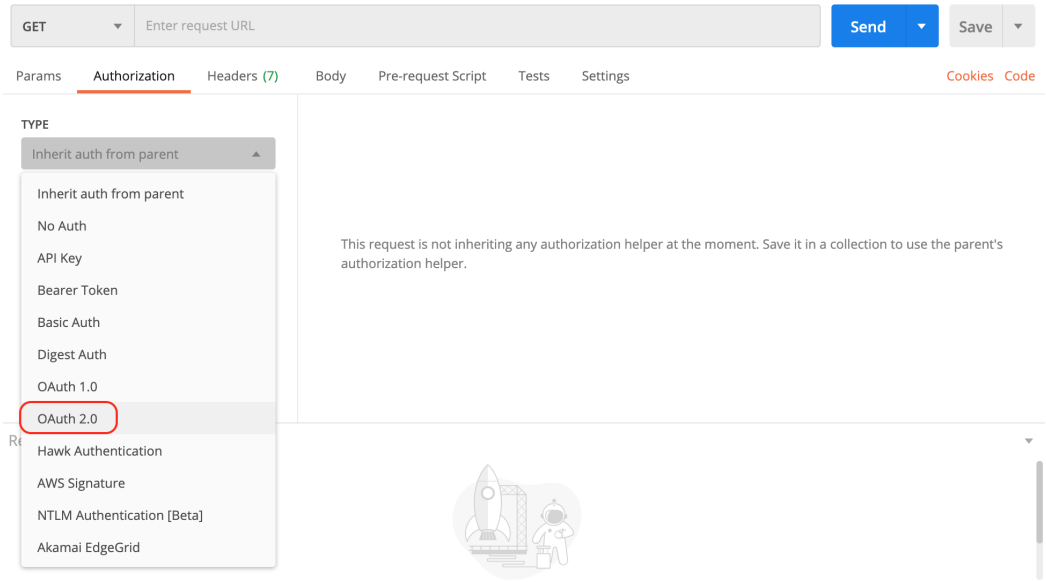
- When the input is complete, please click the `Configure` button



- 5.4 Set `Authorization` in Postman
  - Download `Postman` and Install
    - https://www.postman.com/downloads/
  - Open `Postman` and Click `Authorization` tab

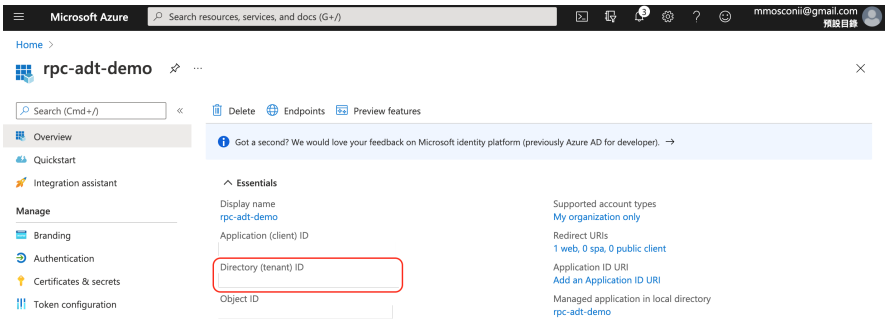


  - `Authorization` tab
    - `Type` field
      - Select `OAuth 2.0`

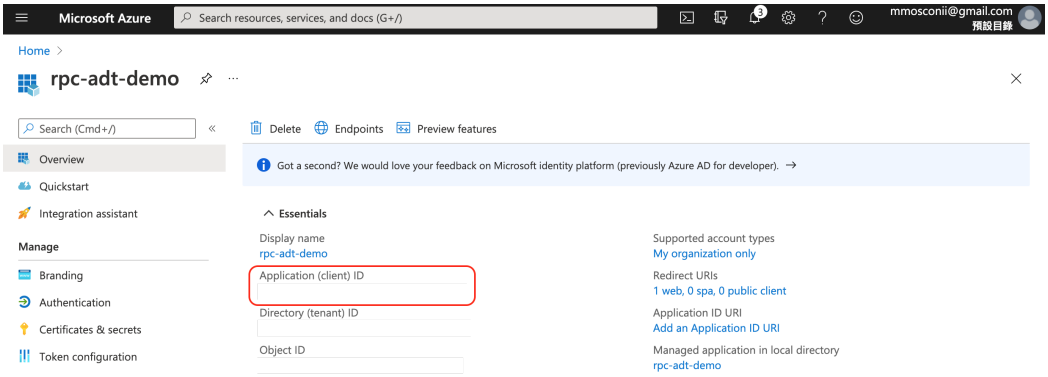


    - `Configure New Token`
      - `Grant Type` field

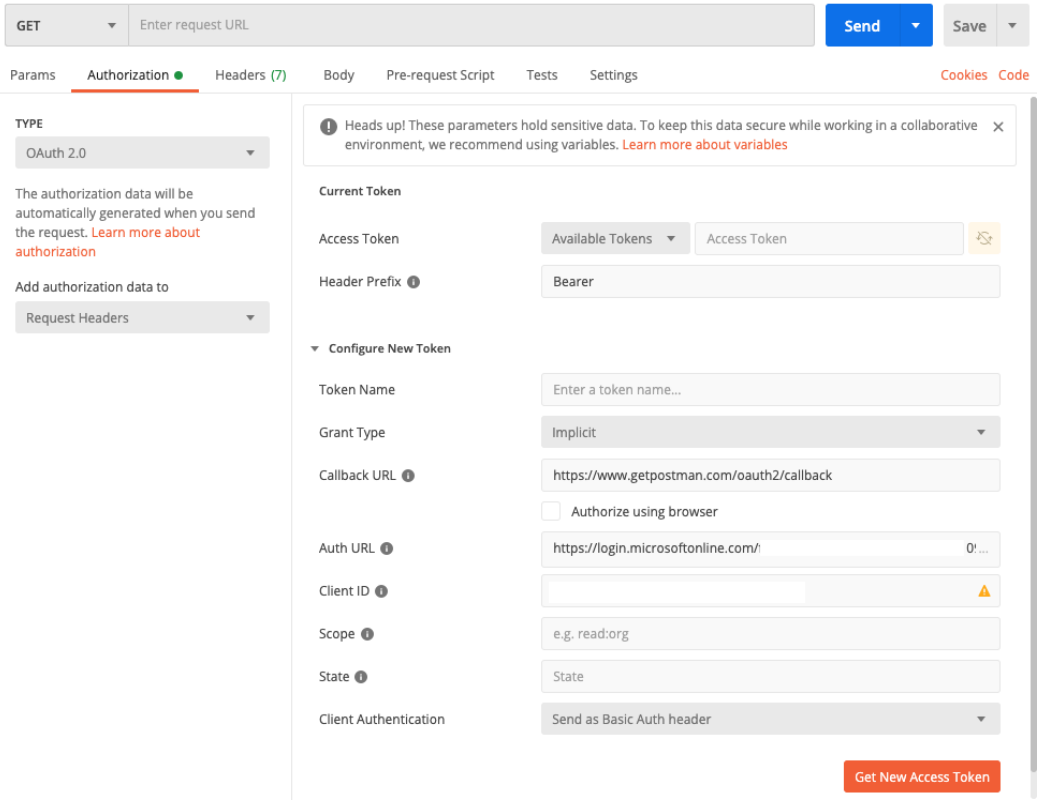
    - Select `Implicit`
- `Callback URL` field
    - Please input `https://www.getpostman.com/oauth2/callback`
- `Auth URL` field
    - https://login.microsoftonline.com/TENANT_ID/oauth2/authorize?resource=0b07f429-9f4b-4714-9392-cc5e8e80c8b0
        - Modify `TENANT_ID` to `Directory (tenant) ID`



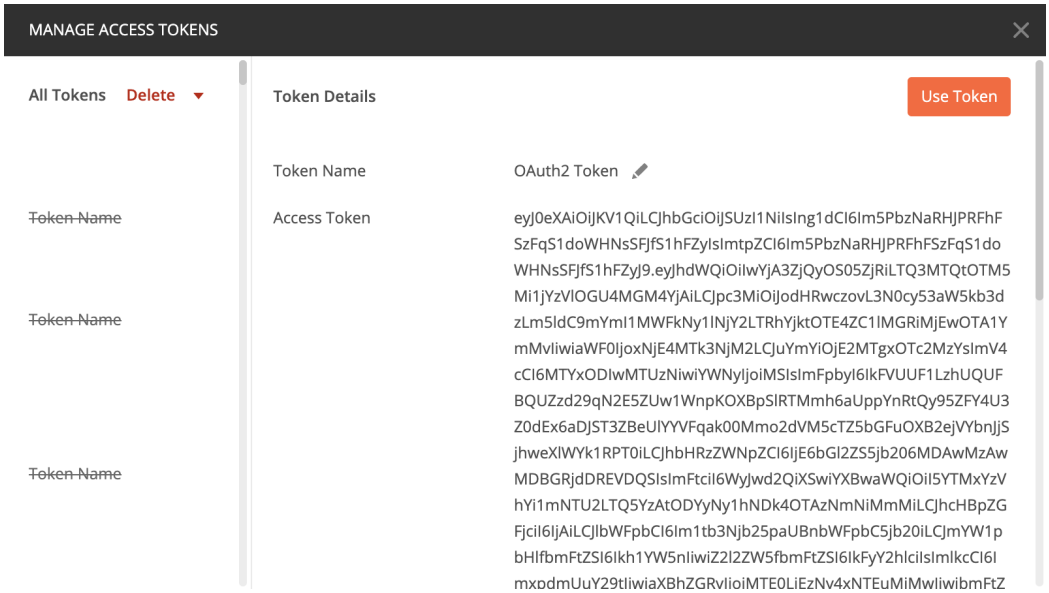
- `Client ID` field



- When the input is complete, please click the `Get New Access Token` button

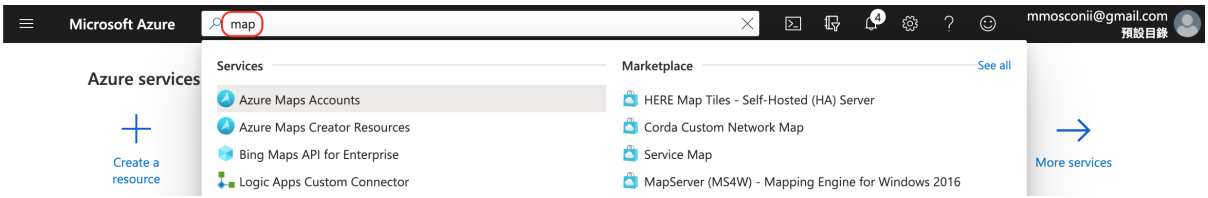


- Click `Use Token`

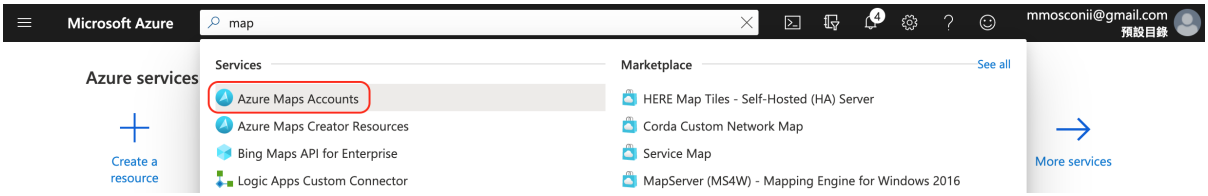


6. Upload DWG & manifest.json to Azure Map using Postman

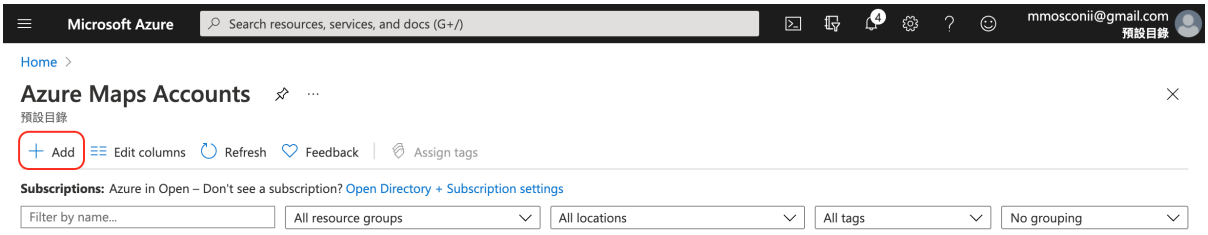
- 6.1 Create `Azure Maps Account`
  - 6.1.1 Search `map`



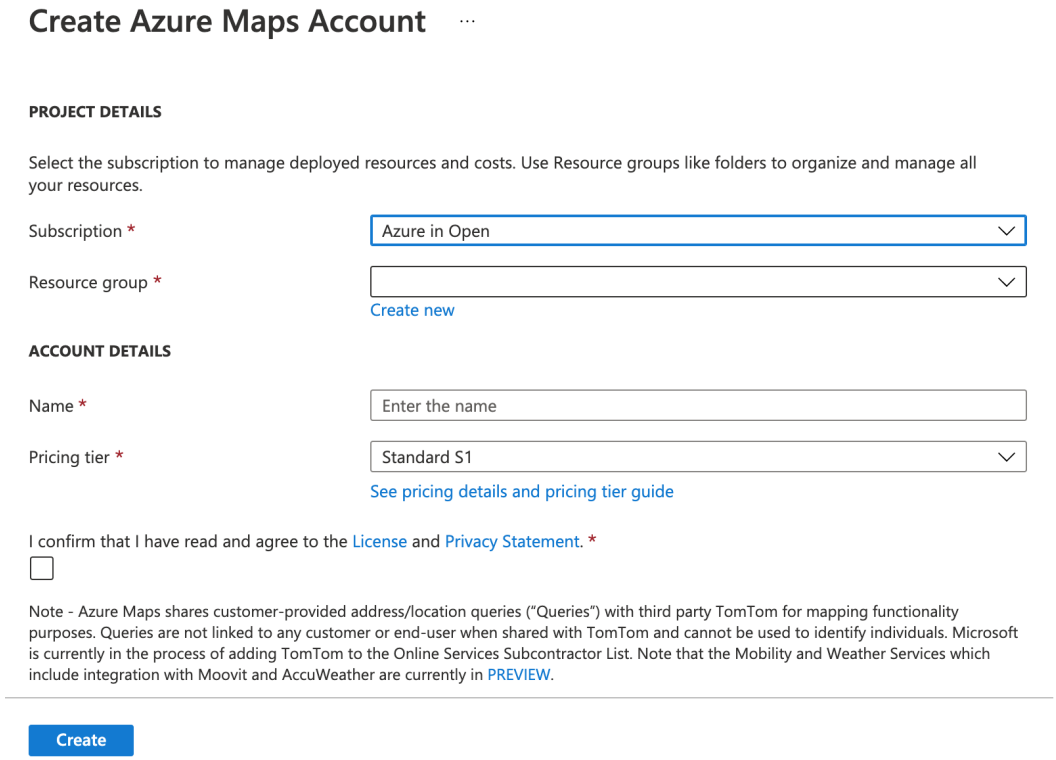
  - 6.1.2 Select `Azure Maps Accounts`



  - 6.1.3 Click `Add`

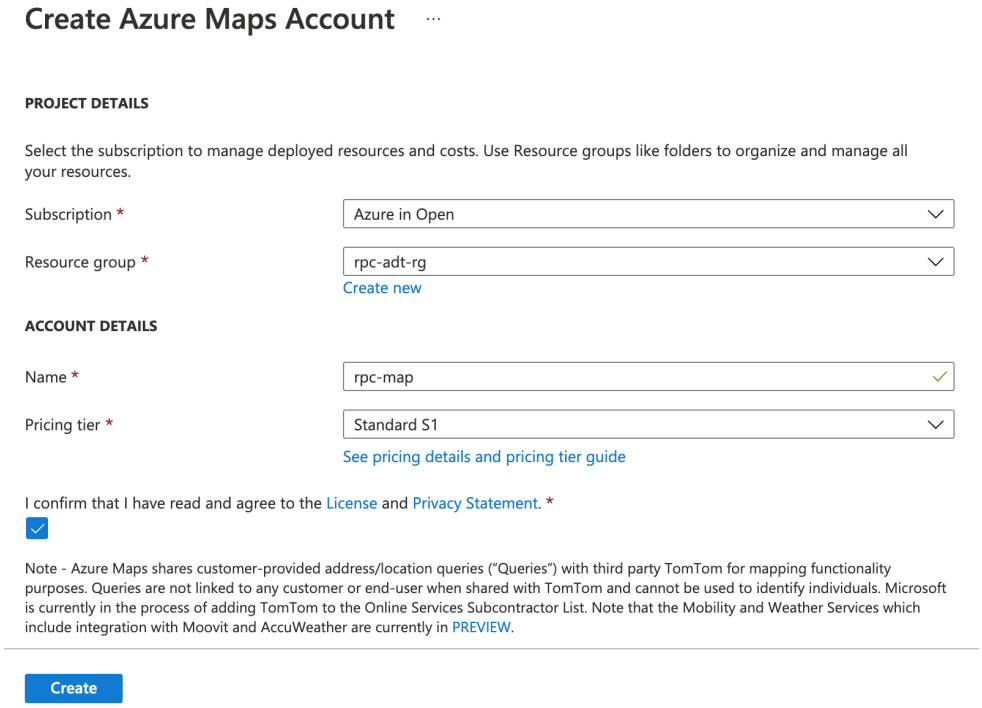


- 6.1.4 Input data

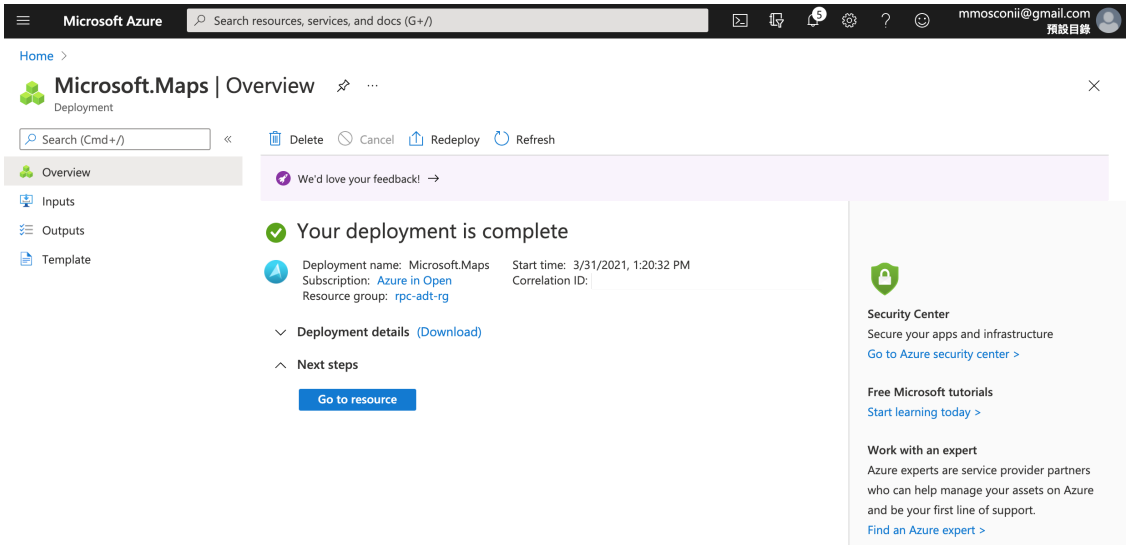


  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name, this example uses `rpc-adt-rg`
  - `Name` field
    - Please use a recognizable name, this example uses `rpc-map`
  - `Pricing tier` field
    - this example uses `Standard S1`
  - `I confirm that I have read and agree to the License and Privacy Statement.` field
    - Please click `check`

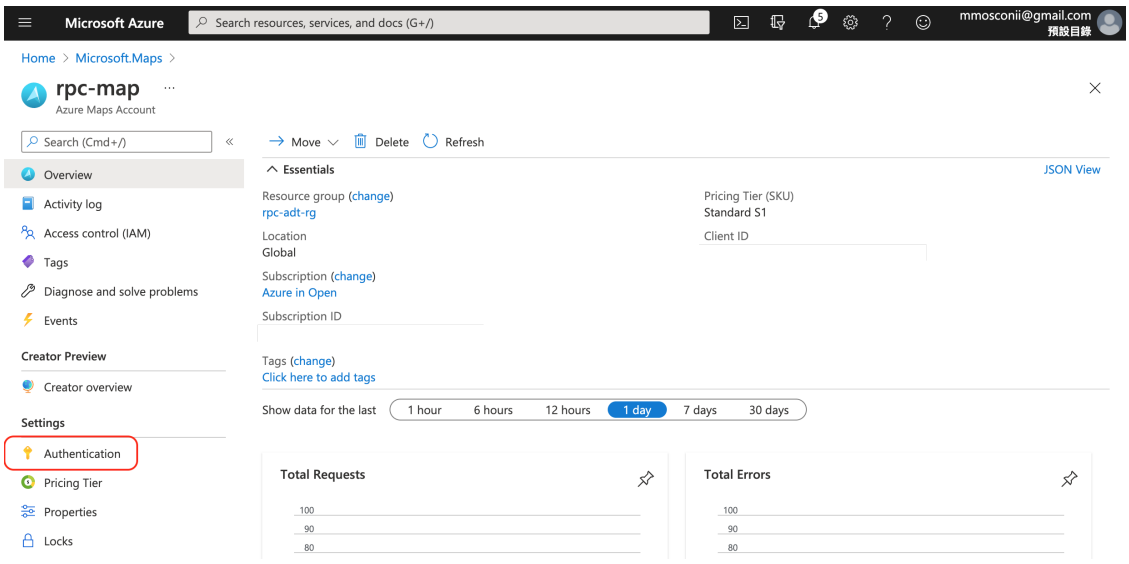
- When the input is complete, please click the `Create` button



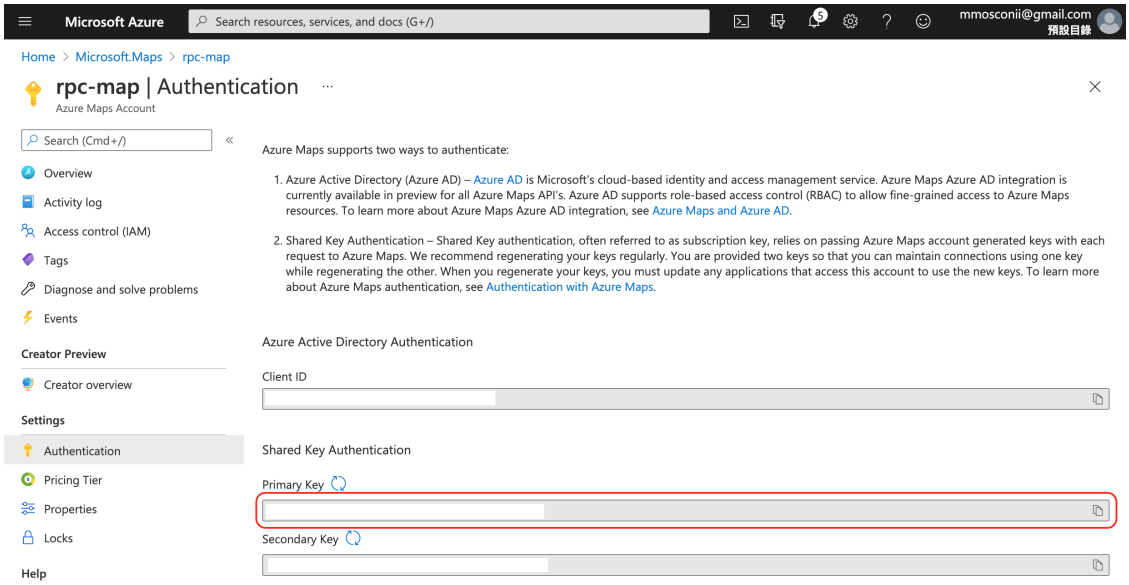
- Once deployment complete click on `Go to resource` button



- 6.1.5 Get `subscription-key`
  - Click `Authentication`

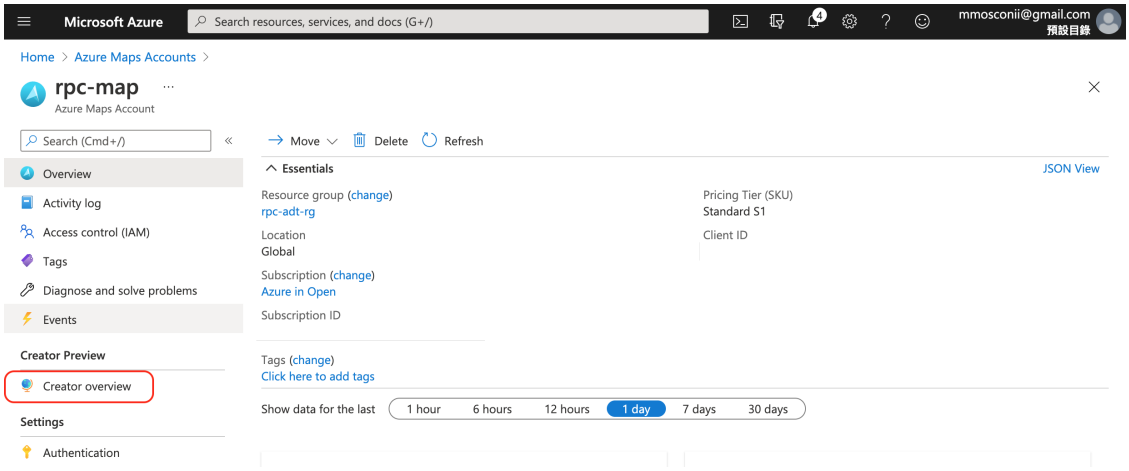


- Get `Primary Key`

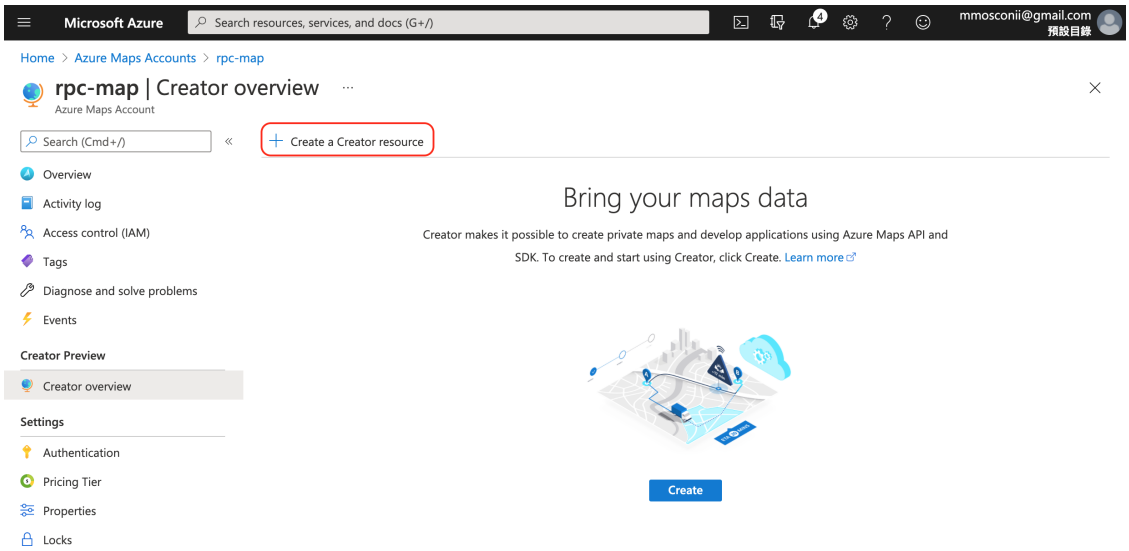


6.1.6 Create `Creator overview`

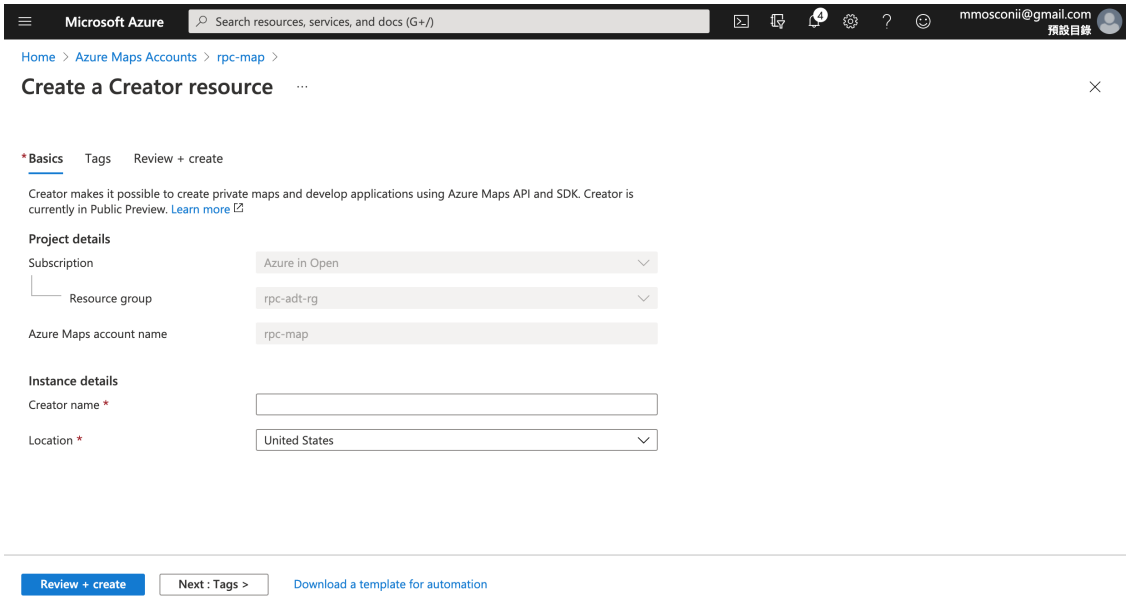
- Click `Creator overview`



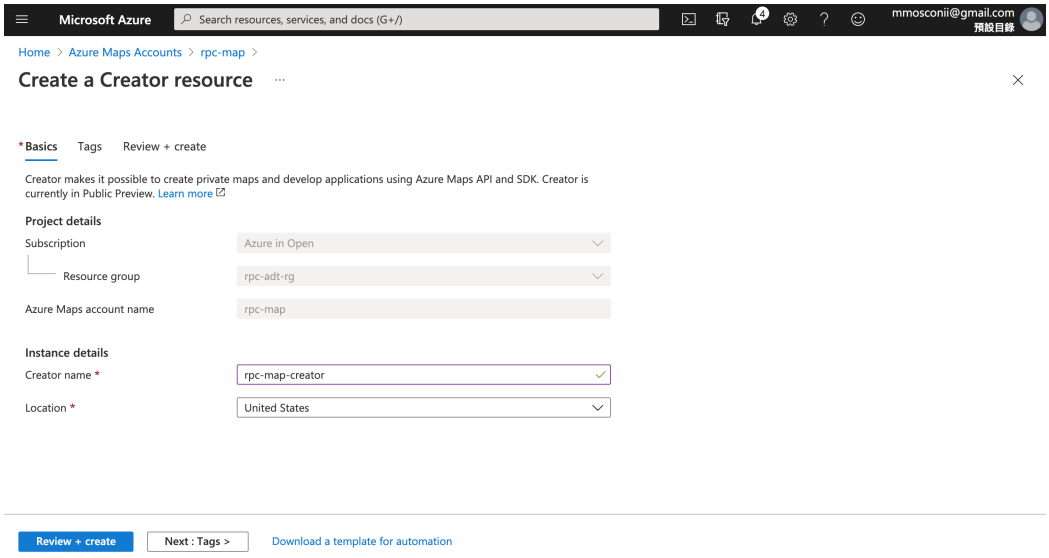
- Click `+ Create a Creator resource`



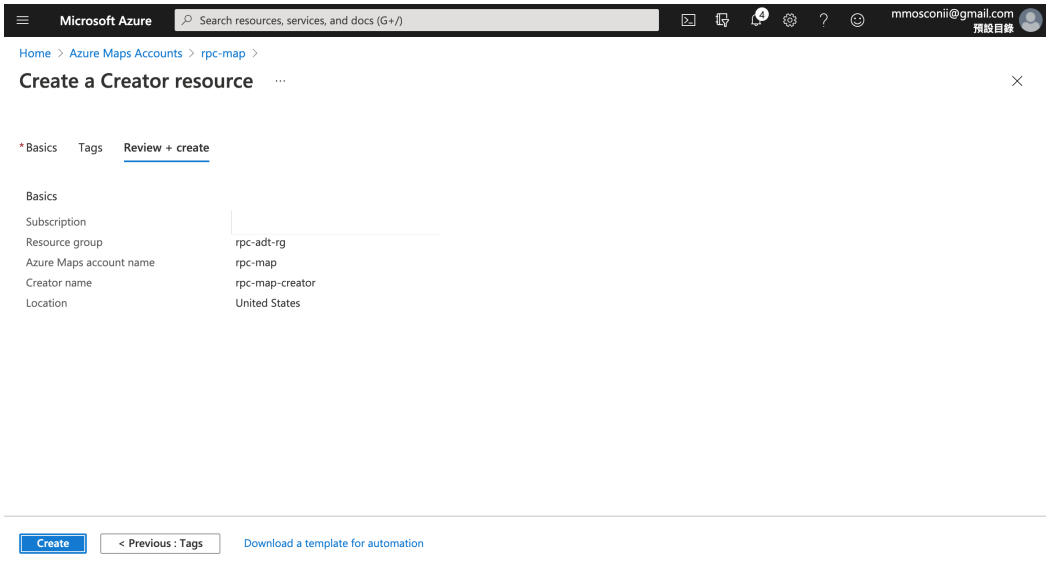
- Input data



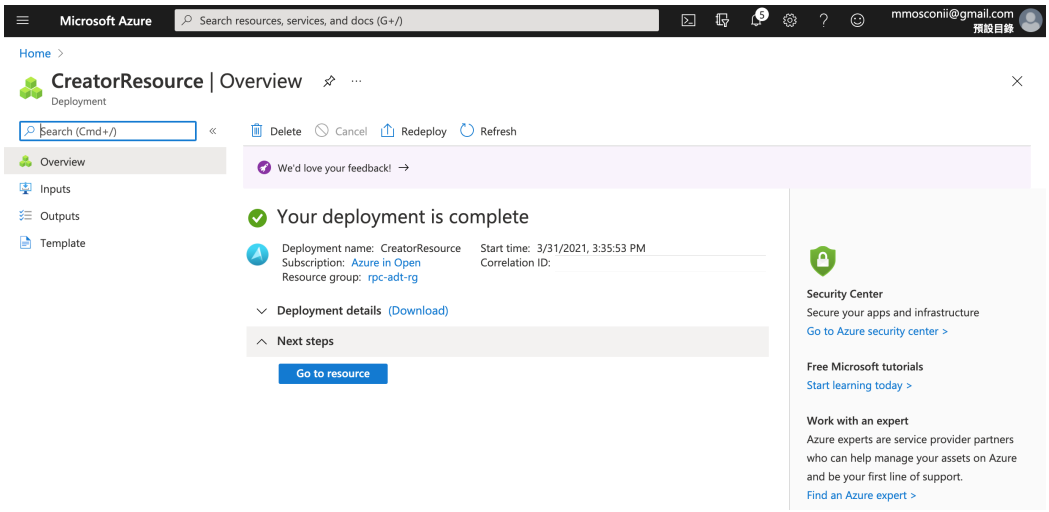
  - `Creator name` field
    - Please use a recognizable name, this example uses `rpc-map-creator`
  - When the input is complete, please click the `Review + create` button



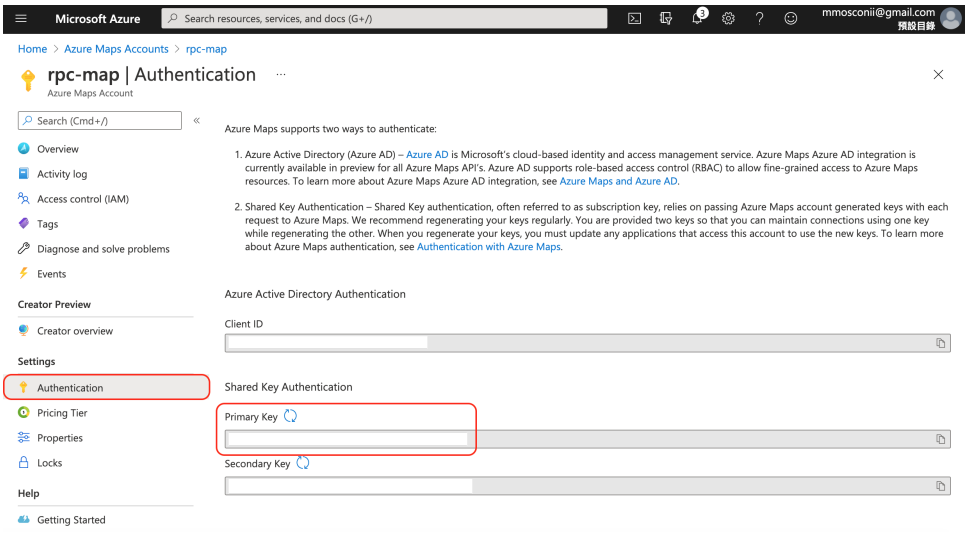
  - Review your settings and select Create



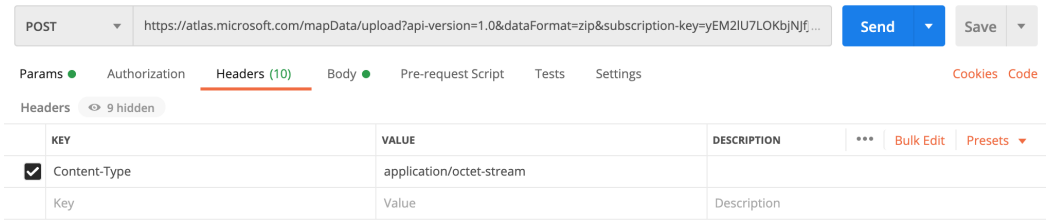
    - When the deployment completes, you'll see a page with a success or a failure message



- 6.2 Download and install `Postman`
    - https://www.postman.com/downloads/
- 6.3 Upload a Drawing package to `Azure Map`
    - 6.3.1 Create Request
        - HTTP Method
            - POST
        - Request
            - https://atlas.microsoft.com/mapData/upload?api-version=1.0&dataFormat=zip&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
            - Comment
                - Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`

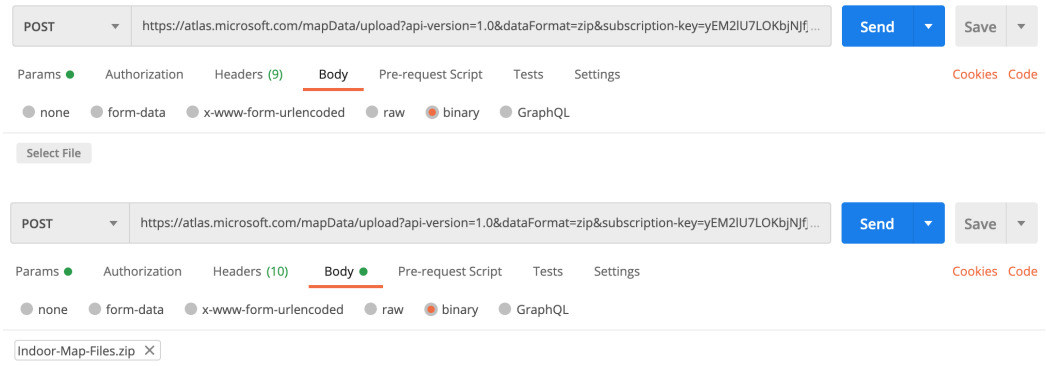


        - Headers
            - Content-Type: `application/octet-stream`

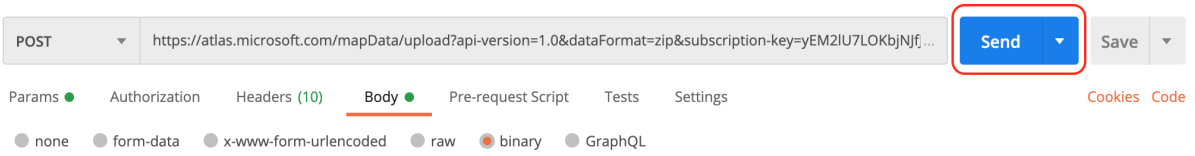


        - Body

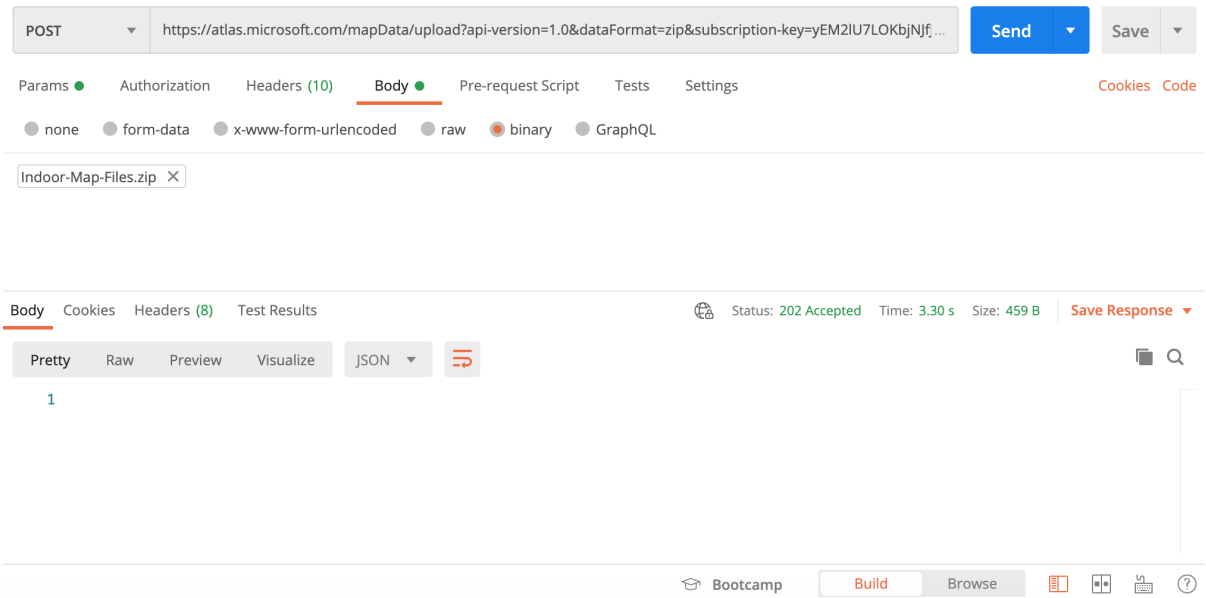
    - Select `binary`
    - Select File: Upload `./Azure-Digital-Twins-for-RPC/Indoor-Map-Files/Indoor-Map-Files.zip`



- 6.3.2 Click the blue `Send` button and wait for the request to process

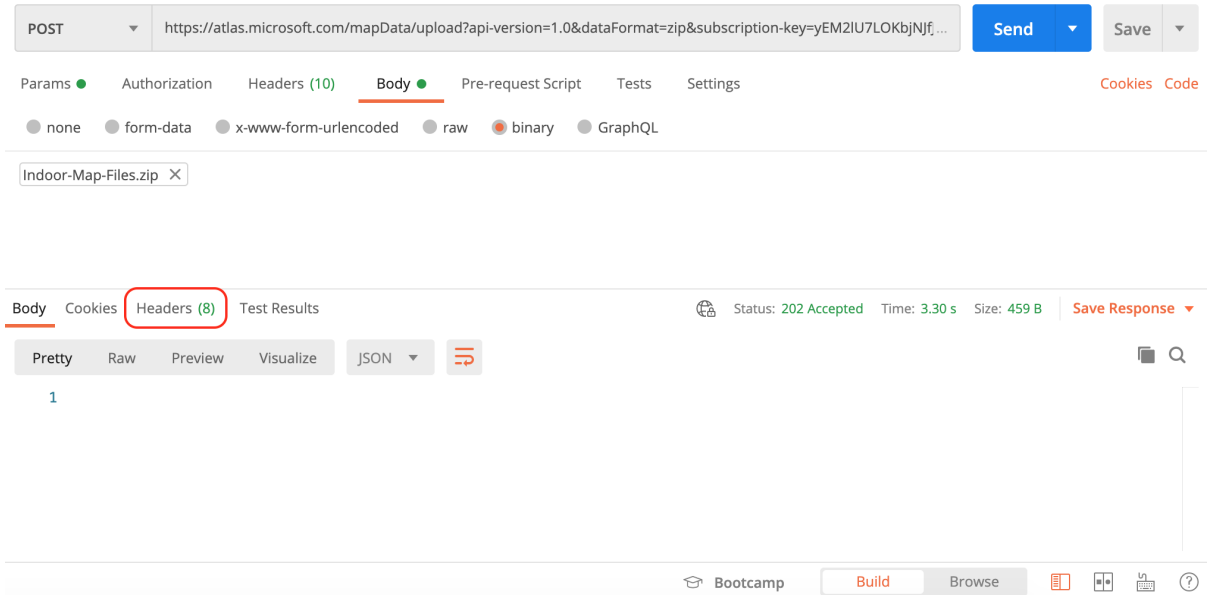


- 6.3.3 Request complete

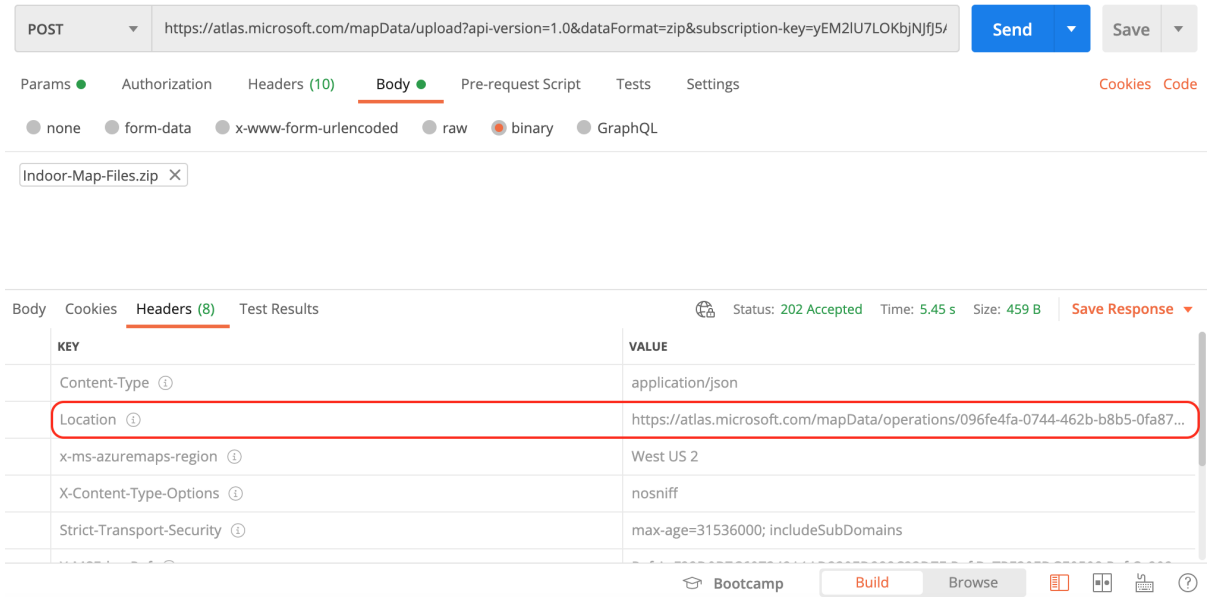


- 6.4 To check the status of the API call

- 6.4.1 Go to the Headers tab of the response

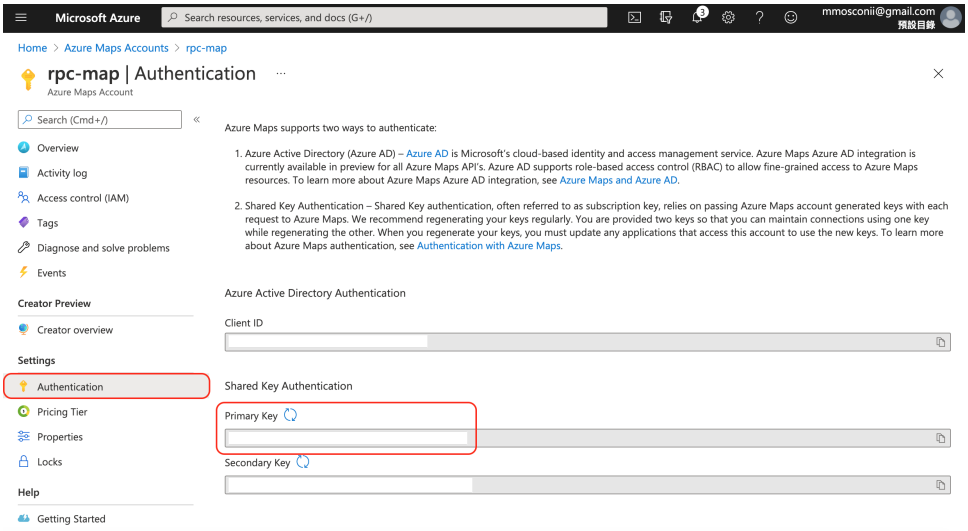


- 6.4.2 Copy the value of the Location key

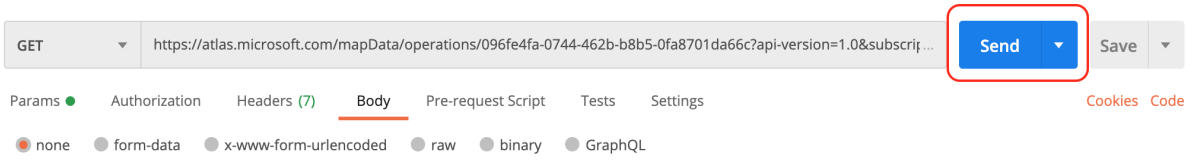


- 6.4.3 Need to append your primary subscription key to the URL for authentication
  - HTTP Method
    - GET
  - Request
    - `https://atlas.microsoft.com/mapData/operations/<operationId>?api-version=1.0&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY`
  - Comment

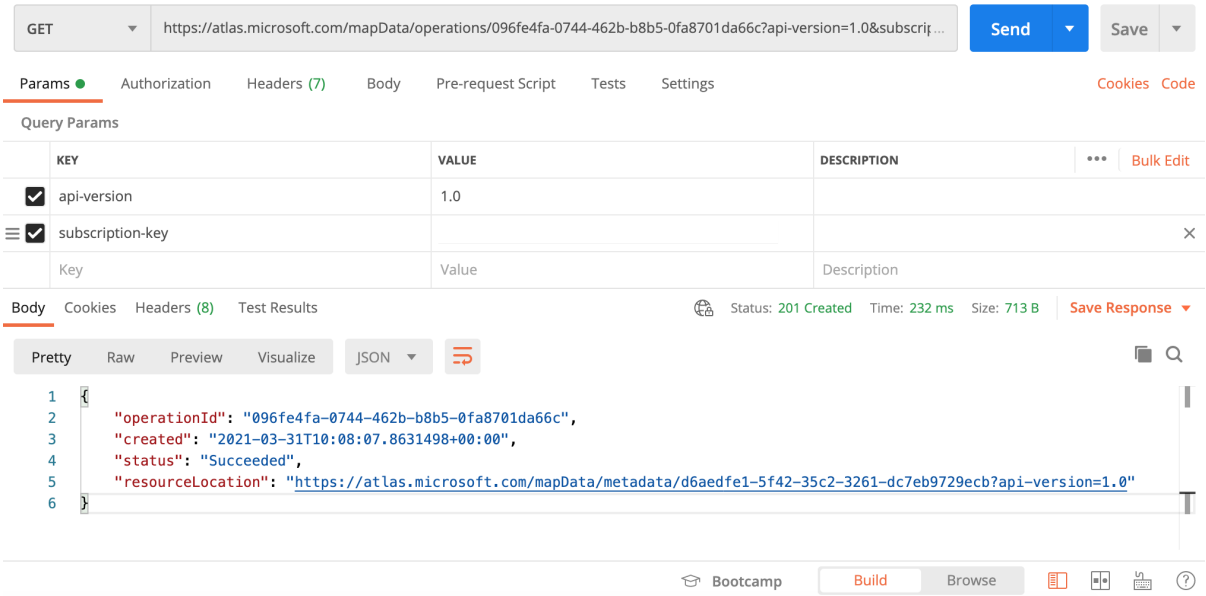
- Modify `AZURE_MAPS_PRIMARY_SUBSCRIPTION_KEY` to `Primary Key`



- 6.4.4 Click the blue `Send` button and wait for the request to process

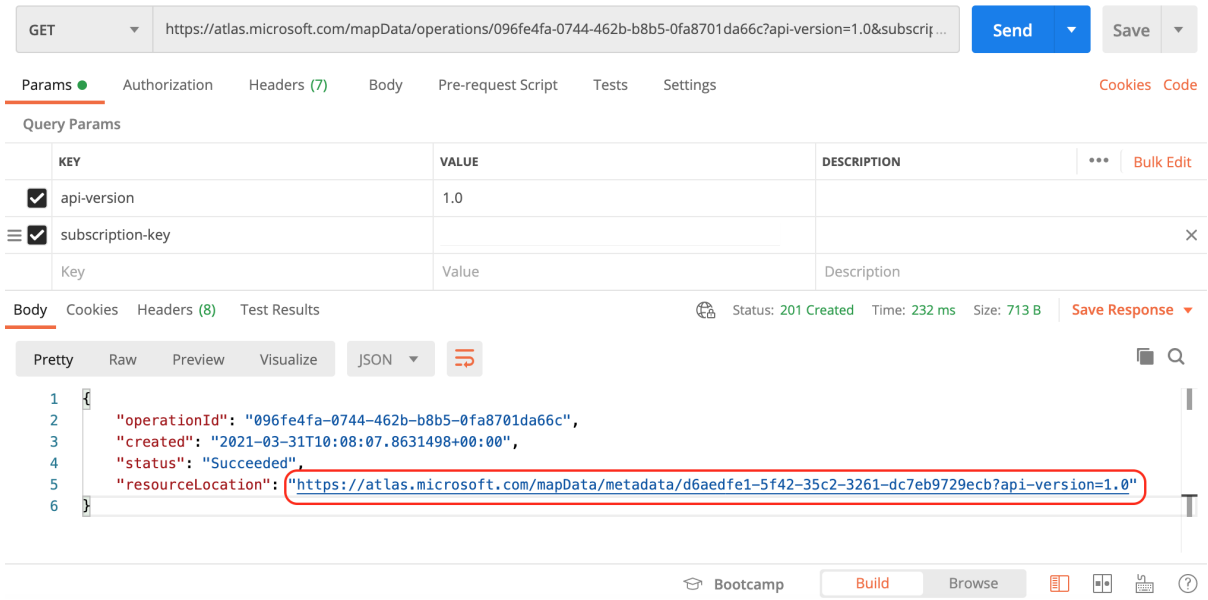


- 6.4.5 Request complete

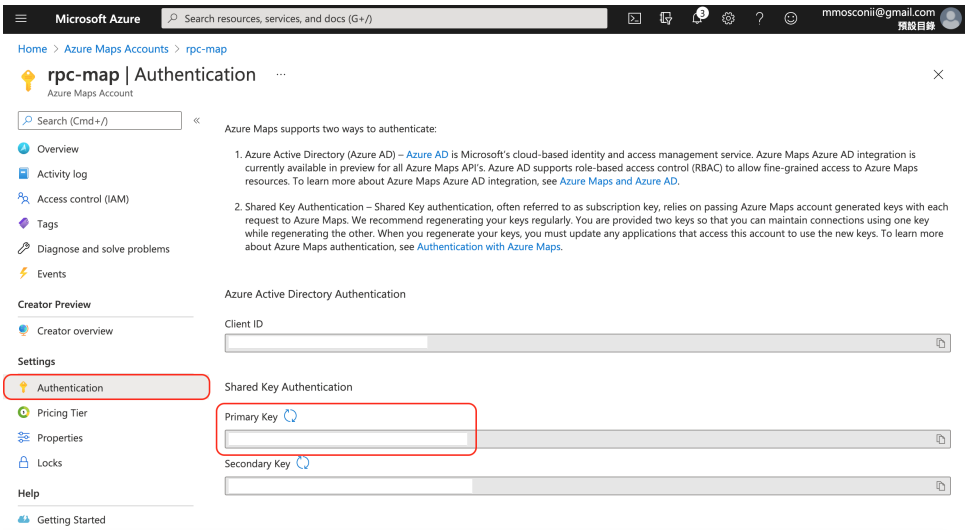


- 6.5 To retrieve content metadata

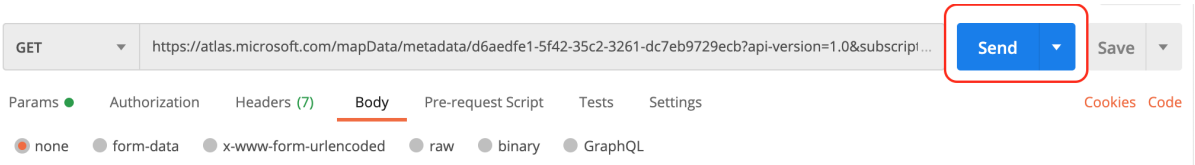
- 6.5.1 Copy the `resourceLocation` that was retrieved in `Step 6.4.5`



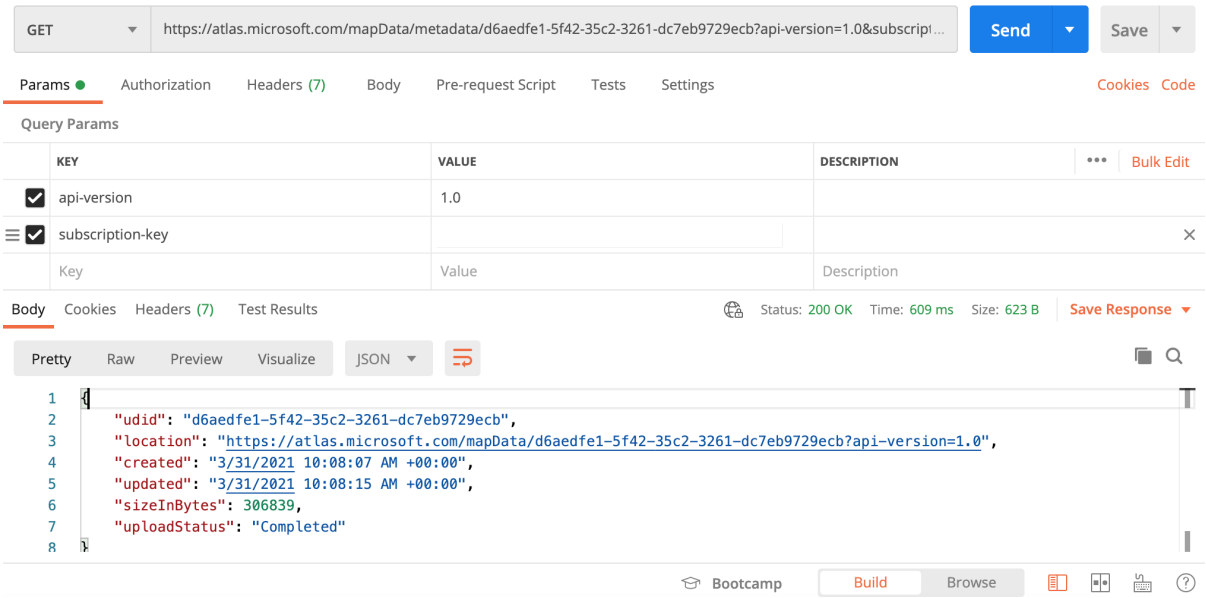
- 6.5.2 Need to append your primary subscription key to the URL for authentication
  - HTTP Method
    - GET
  - Request
    - https://atlas.microsoft.com/mapData/metadata/{udid}?api-version=1.0&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
    - Comment
      - Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



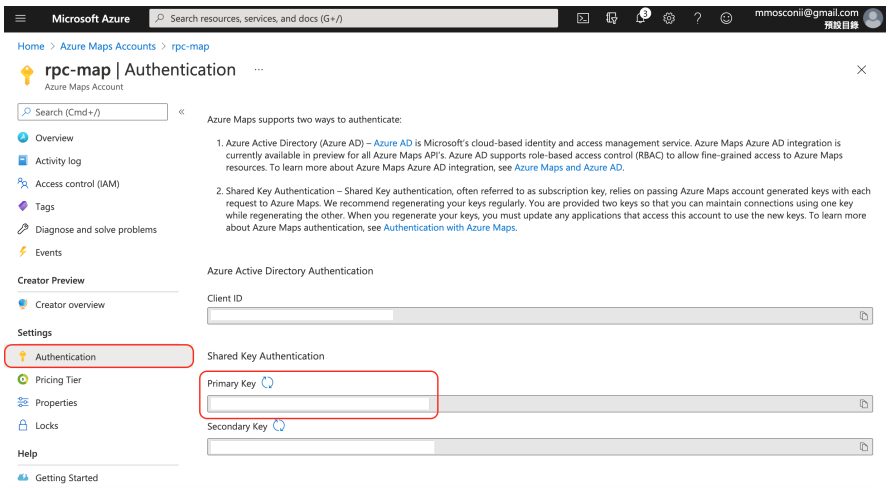
- 6.5.3 Click the blue `Send` button and wait for the request to process



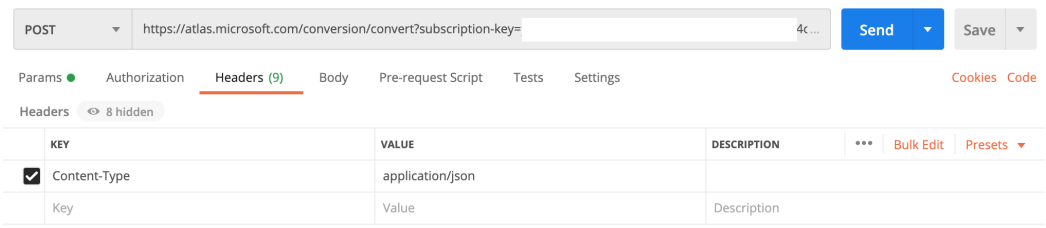
    - 6.5.4 Request complete



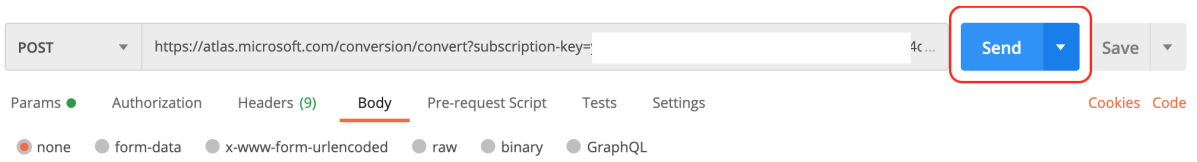
- 6.6 Convert a Drawing package
    - 6.6.1 Create convert request
        - HTTP Method
            - POST
        - Request
            - https://atlas.microsoft.com/conversion/convert?subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY&api-version=1.0&udid=UDID&inputType=DWG
            - Comment
                - 1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



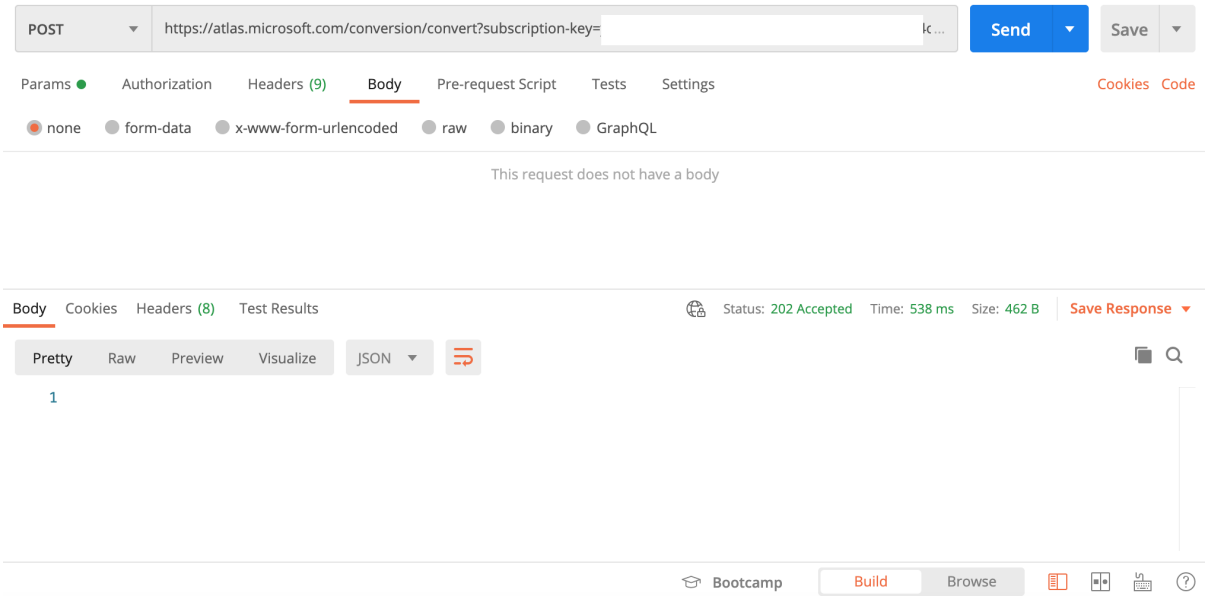
                - 2. Modify `UDID` to `udid` that was retrieved in `Step 6.5.2`
        - Headers
            - Content-Type: `application/json`



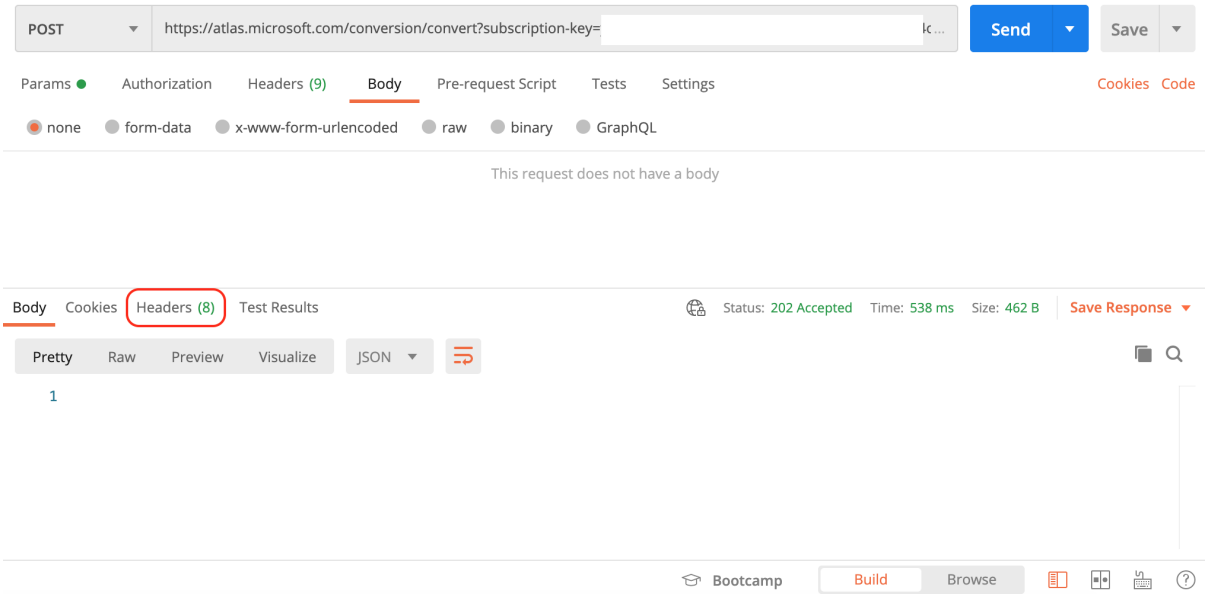
- 6.6.2 Click the blue Send button and wait for the request to process



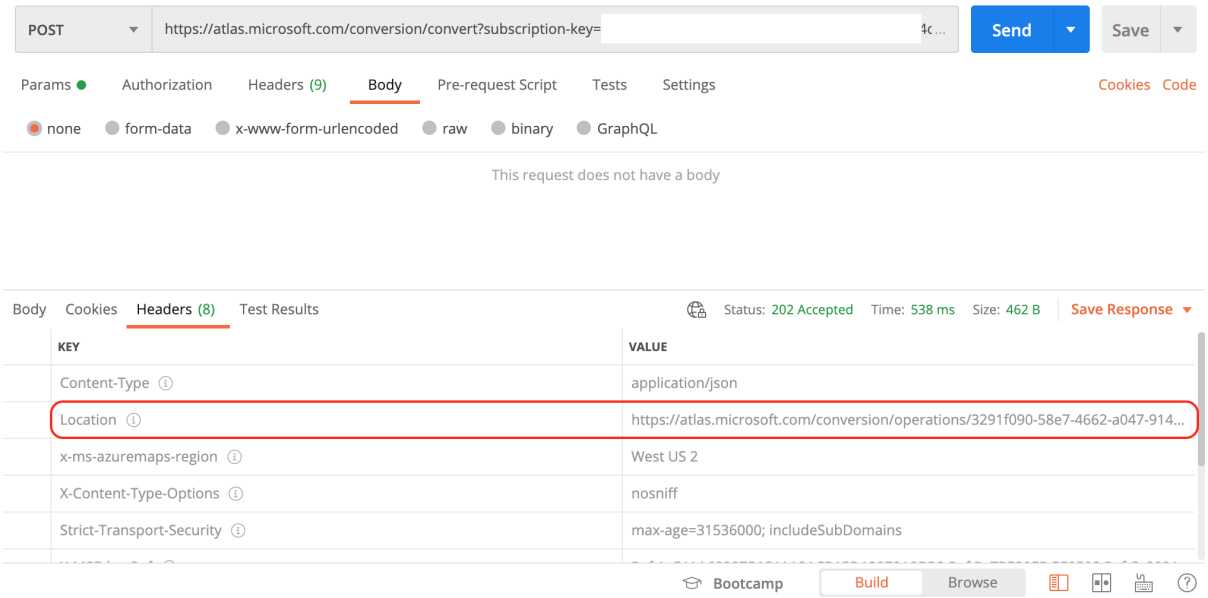
- 6.6.3 Request complete



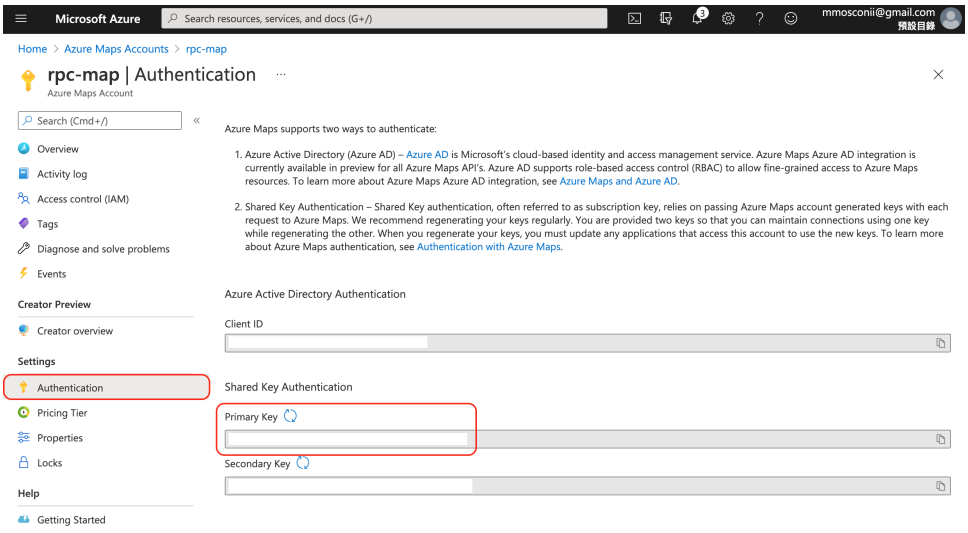
- 6.6.4 Go to the Headers tab of the response, and look for the Location key



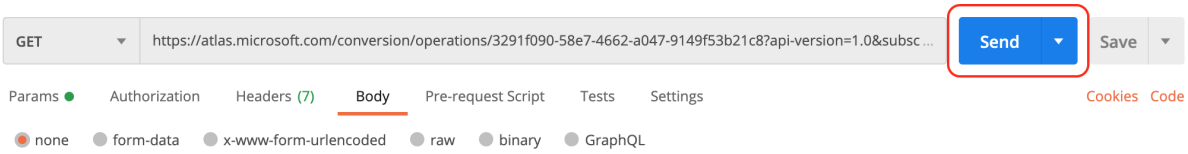
- 6.6.5 Copy the value of the Location key



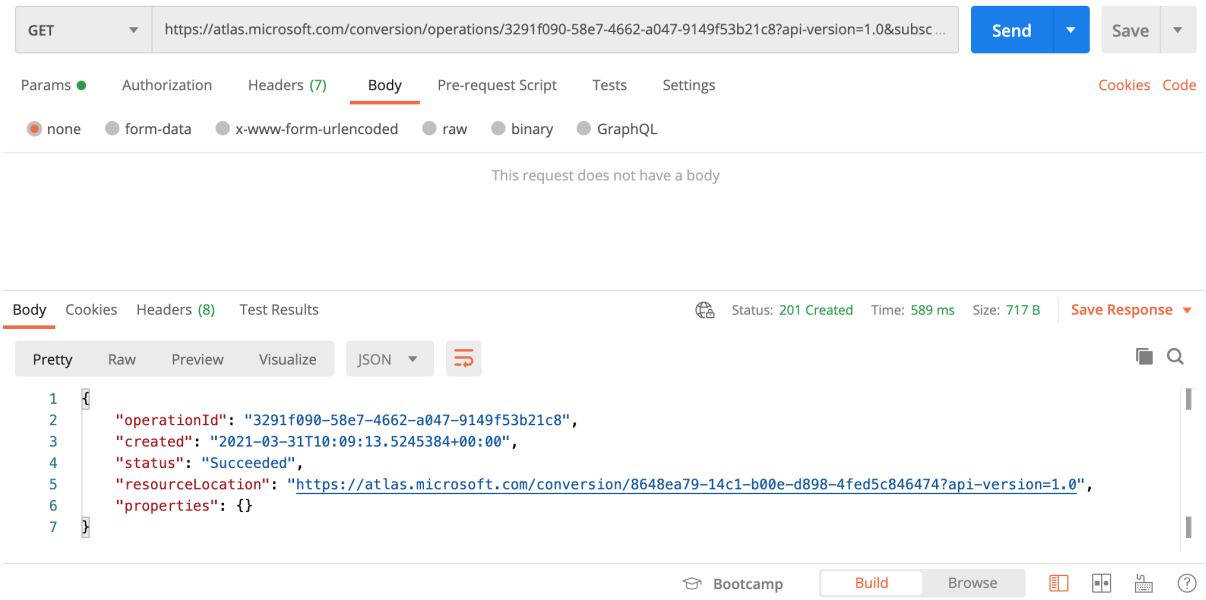
- 6.6.6 Need to append your primary subscription key to the URL for authentication
  - HTTP Method
    - GET
  - Request
    - `https://atlas.microsoft.com/conversion/operations/<operationId>?api-version=1.0&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY`
    - Comment
      - Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



- 6.6.7 Click the blue `Send` button and wait for the request to process

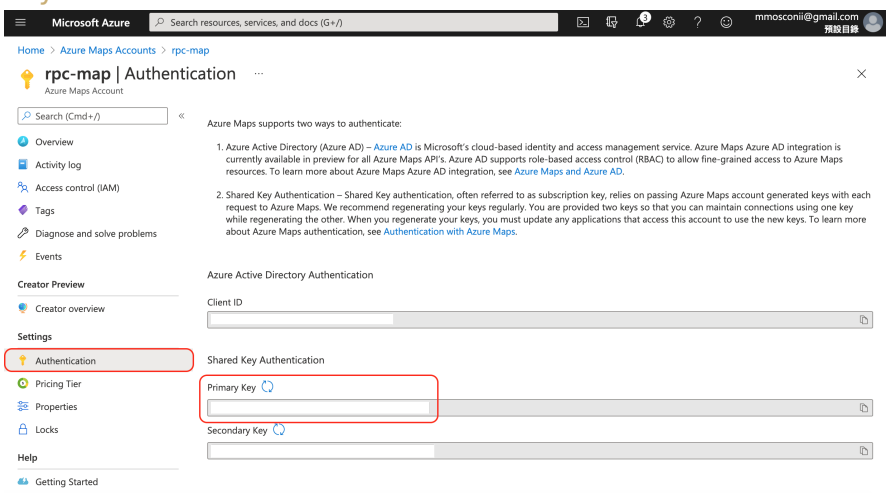


- 6.6.8 Request complete



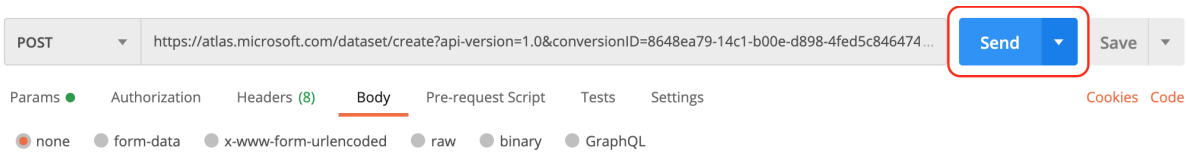
  - Comment
    - Copy the conversionId from the resourceLocation URL for the converted package. The conversionId is used by other API to access the converted map data.
      - https://atlas.microsoft.com/conversion/8648ea79-14c1-b00e-d898-4fed5c846474?api-version=1.0
      - https://atlas.microsoft.com/conversion/CONVERSIONID?api-version=1.0

- 6.7 Create a dataset
  - 6.7.1 Create dataset request
    - HTTP Method
      - POST
    - Request
      - https://atlas.microsoft.com/dataset/create?api-version=1.0&conversionID=CONVERSIONID&type=facility&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
      - Comment
        - 1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`

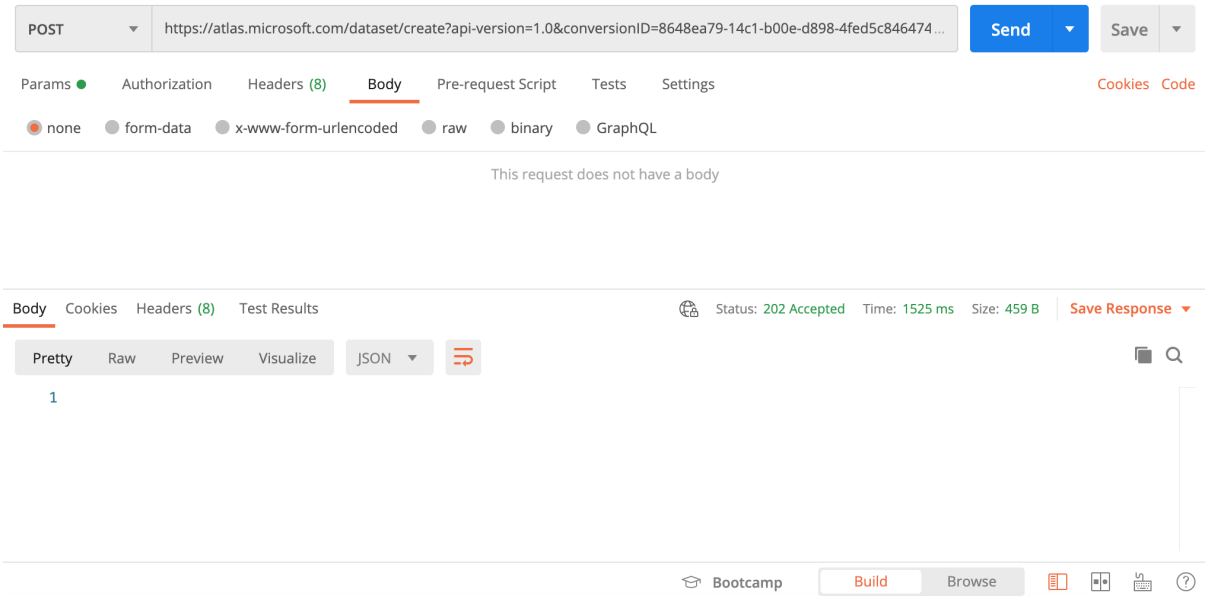


        - 2. Modify `CONVERSIONID` to `CONVERSIONID` that was retrieved in `Step 6.6.8`

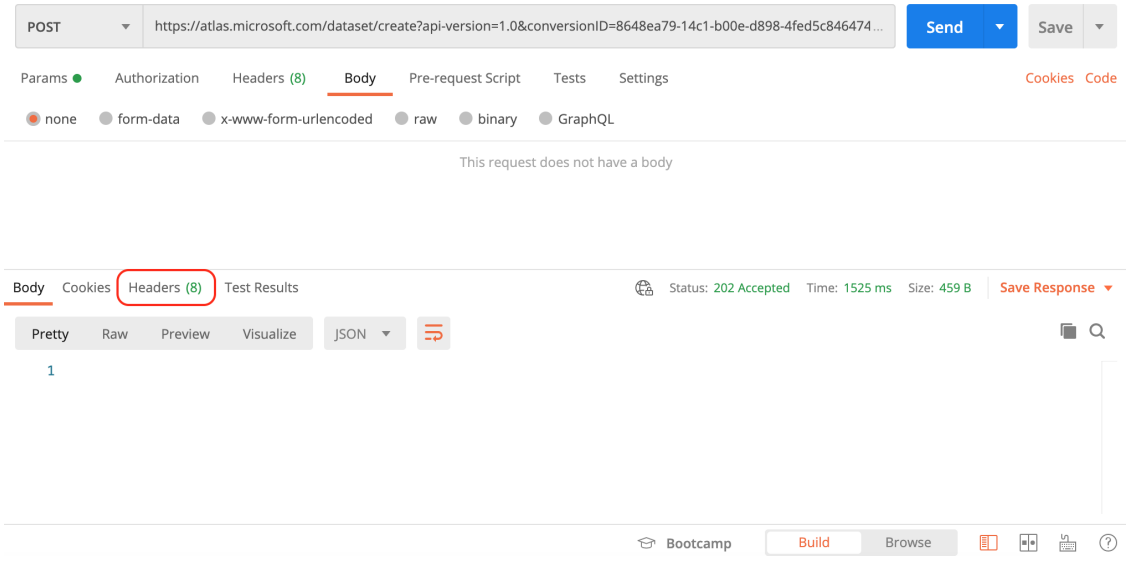
- 6.7.2 Click the blue Send button and wait for the request to process



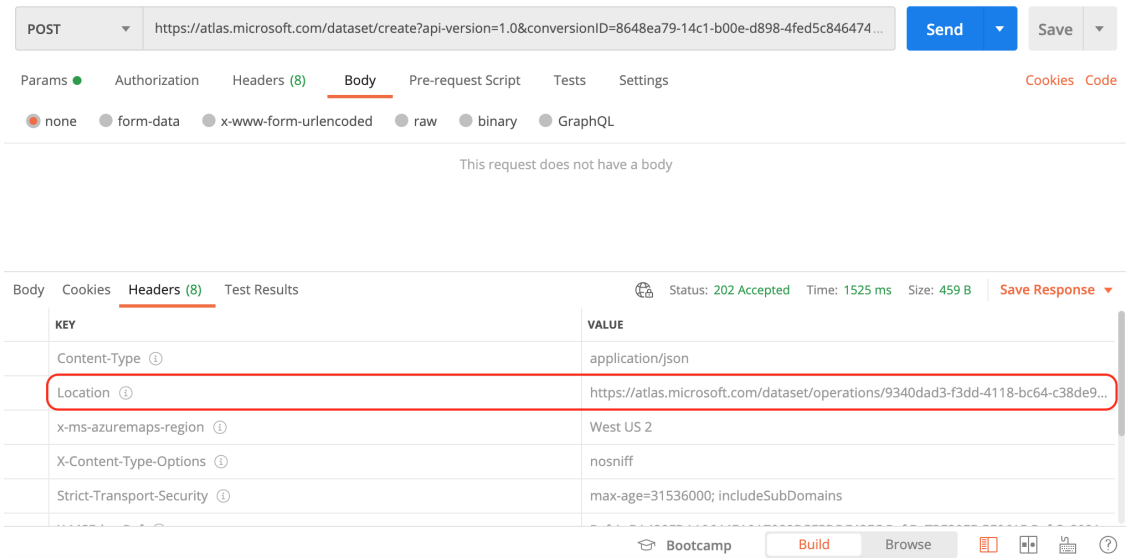
- 6.7.3 Request complete



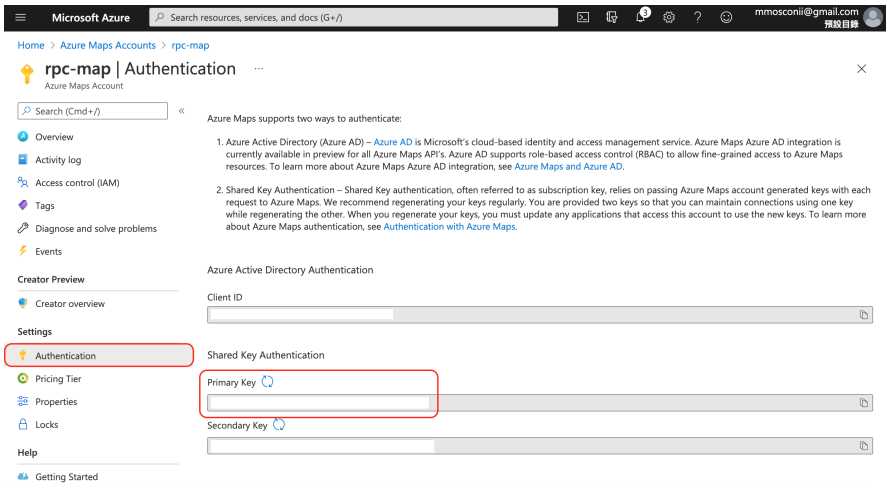
- 6.7.4 Get datasetId
  - Go to the Headers tab of the response



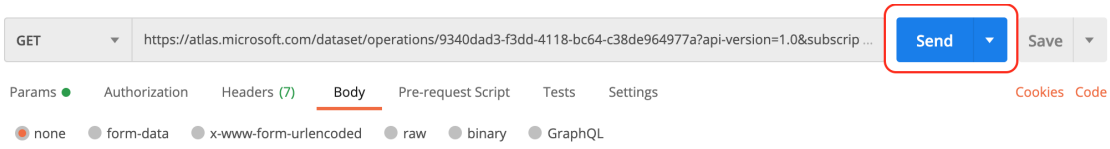
- Copy the value of the Location key



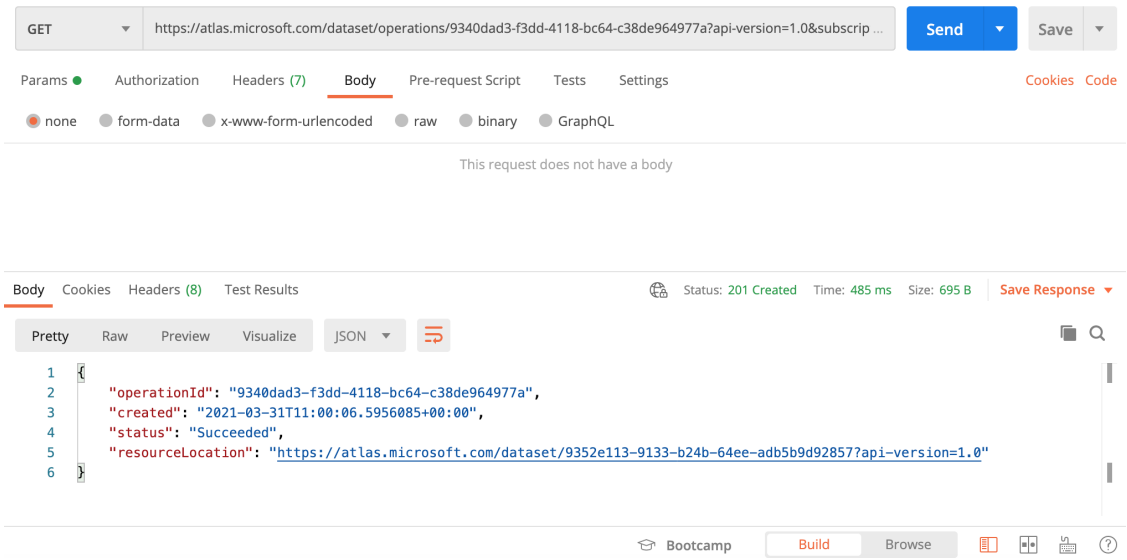
- Need to append your primary subscription key to the URL for authentication
  - HTTP Method
    - GET
  - Request
    - `https://atlas.microsoft.com/dataset/operations/<operationId>?api-version=1.0&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY`
    - Comment
      - Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



- Click the blue `Send` button and wait for the request to process

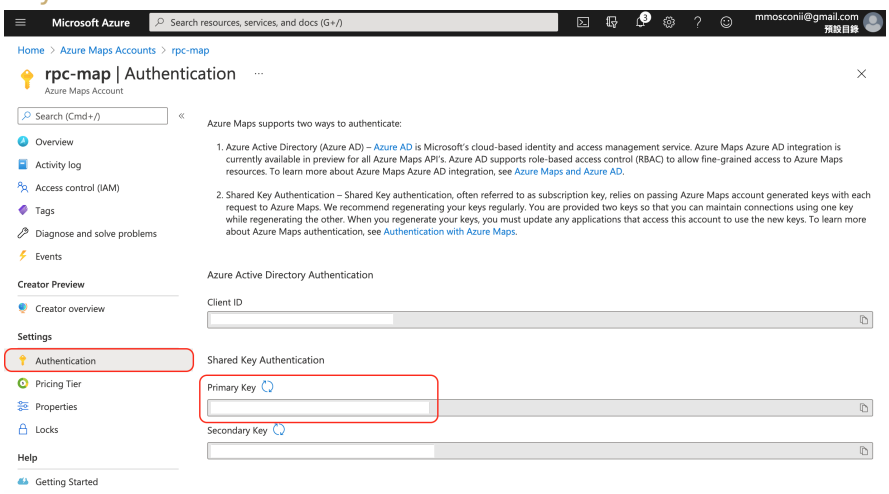


    - Request complete



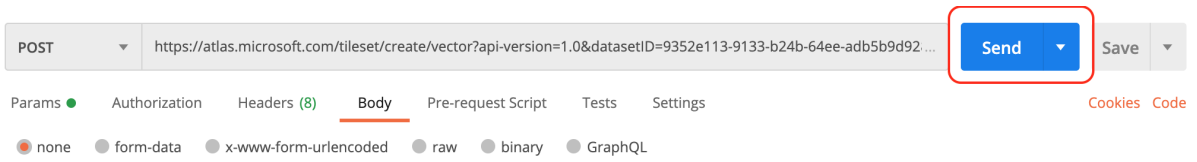
        - Comment
            - The response header will contain the DATASETID for the created dataset. Copy the DATASETID. You'll need to use the DATASETID to create a tileset.
                - `https://atlas.microsoft.com/dataset/9352e113-9133-b24b-64ee-adb5b9d92857?api-version=1.0`
                - `https://azure.microsoft.com/dataset/DATASETID?api-version=1.0`

- 6.8 Create a tileset
    - 6.8.1 Create tileset request
        - HTTP Method
            - POST
        - Request
            - https://atlas.microsoft.com/tileset/create/vector?api-version=1.0&datasetID=DATASETID&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
            - Comment
                - 1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`

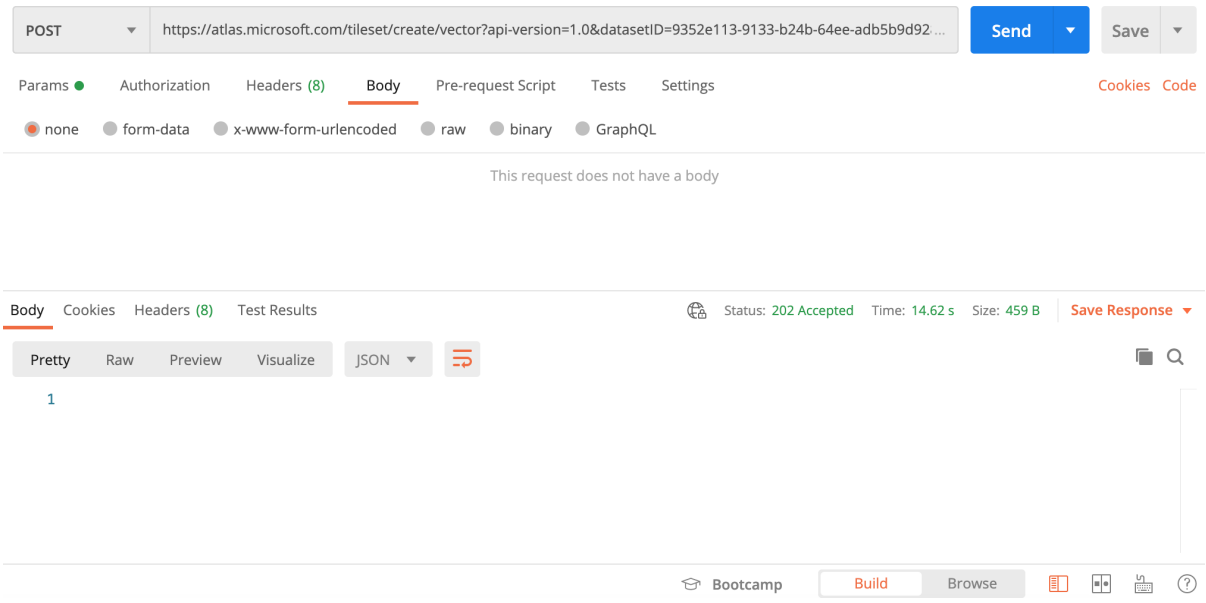


                - 2. Modify `DATASETID` to `DATASETID` that was retrieved in `Step 6.7.4`

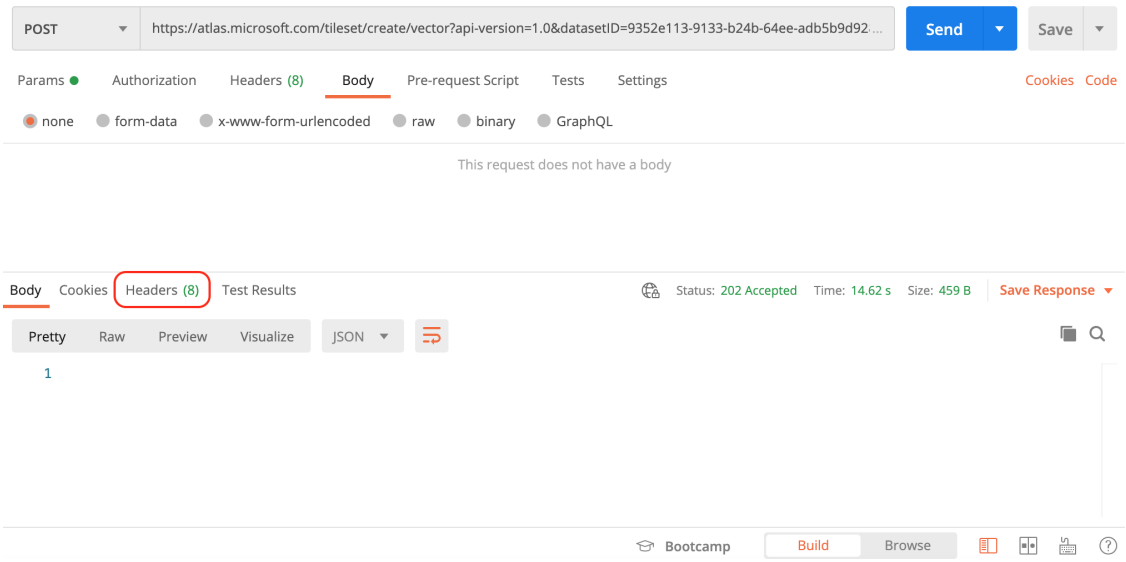
- 6.8.2 Click the blue Send button and wait for the request to process



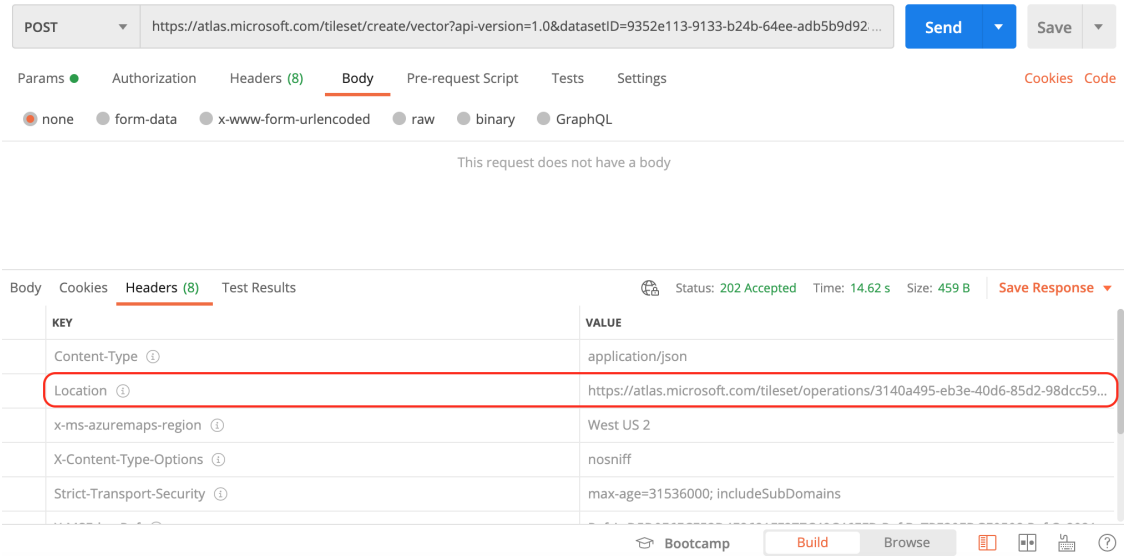
- 6.8.3 Request complete



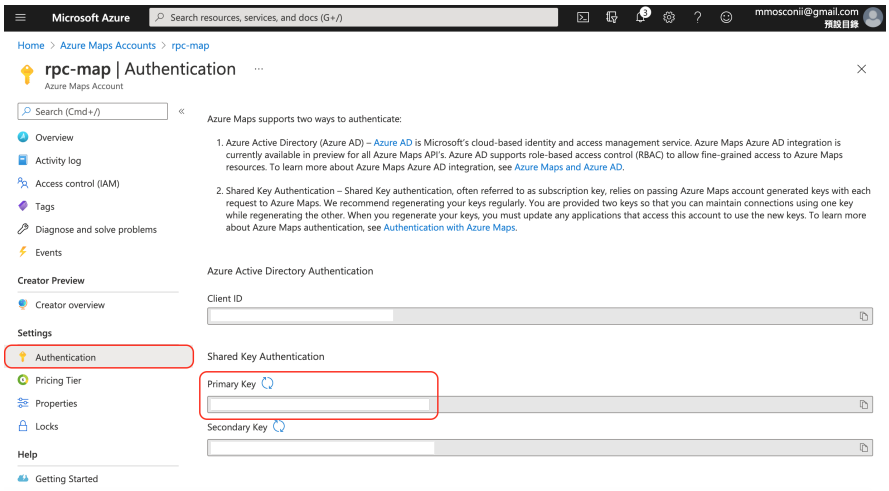
- 6.8.4 Get tilesetId
  - Go to the Headers tab of the response



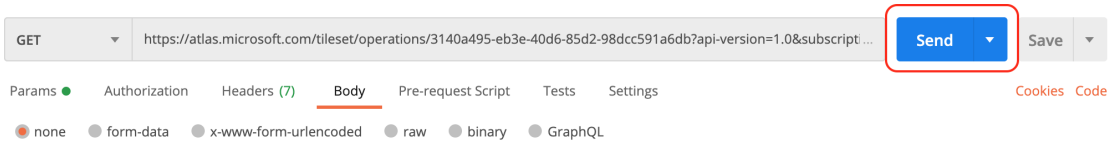
- Copy the value of the Location key



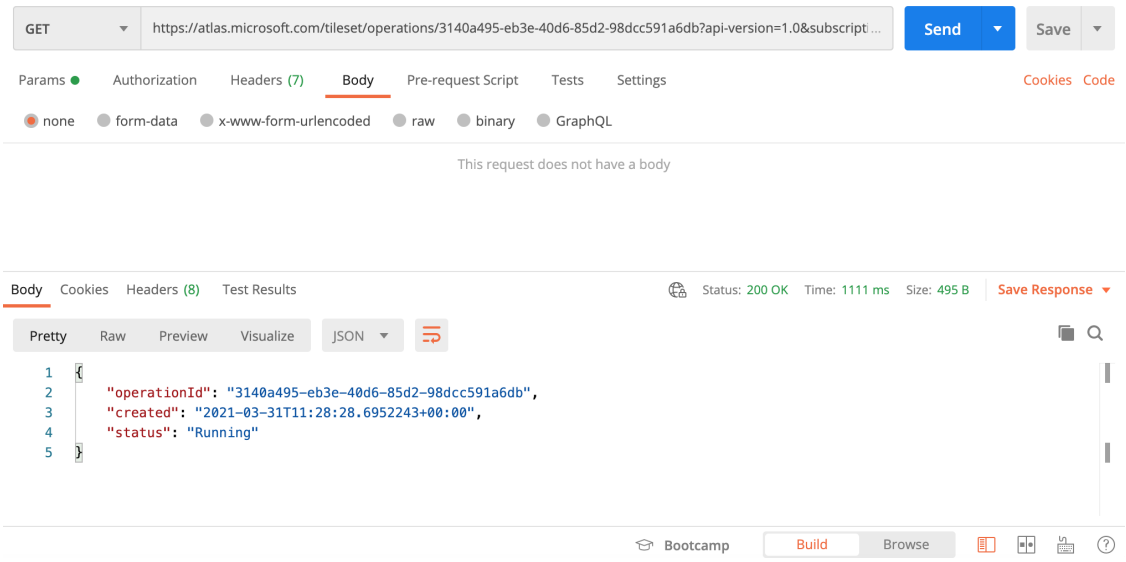
- Need to append your primary subscription key to the URL for authentication
  - HTTP Method
    - GET
  - Request
    - `https://atlas.microsoft.com/tileset/operations/<operationId>?api-version=1.0&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY`
    - Comment
      - Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



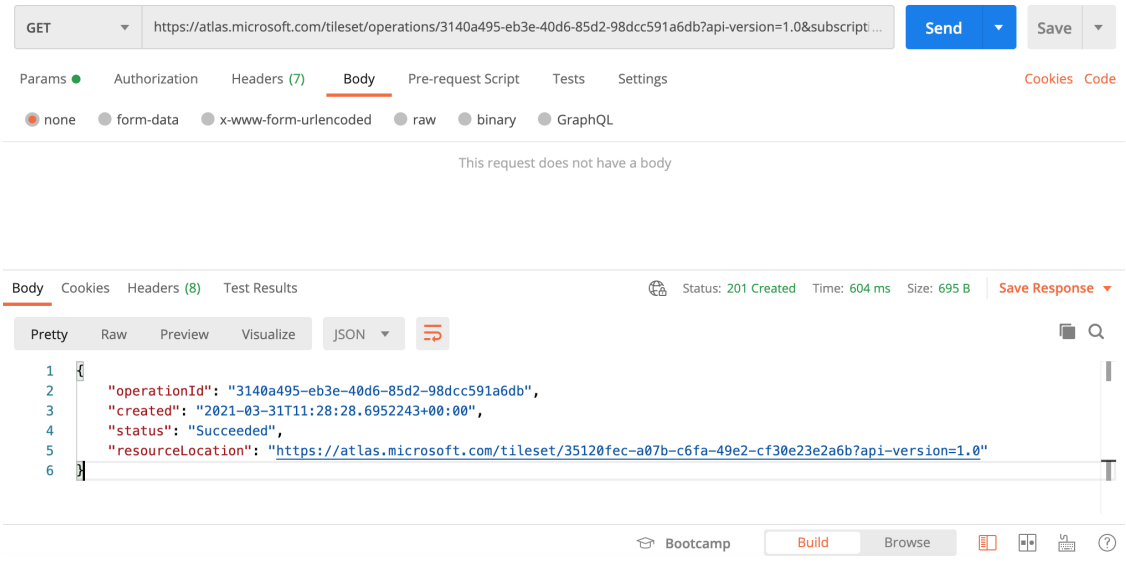
- Click the blue `Send` button and wait for the request to process



- Request running

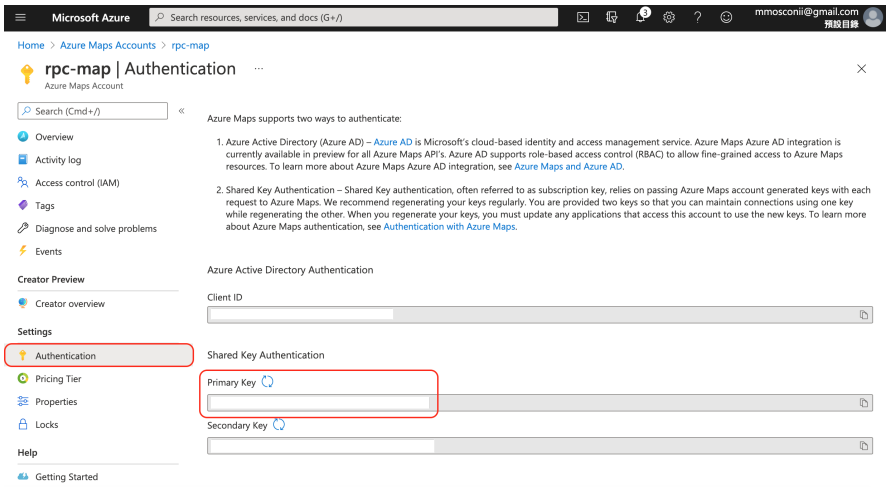


- Request complete

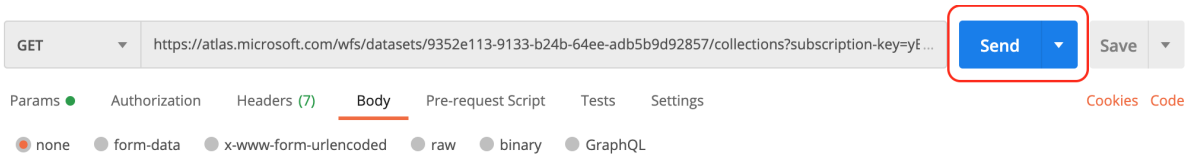


    - Comment
        - The response header will contain the TILESETID for the created dataset. Copy the TILESETID. You'll need to use the TILESETID to create a tileset.
            - `https://atlas.microsoft.com/tileset/35120fec-a07b-c6fa-49e2-cf30e23e2a6b?api-version=1.0`
            - `https://atlas.microsoft.com/tileset/TILESETID?api-version=1.0`

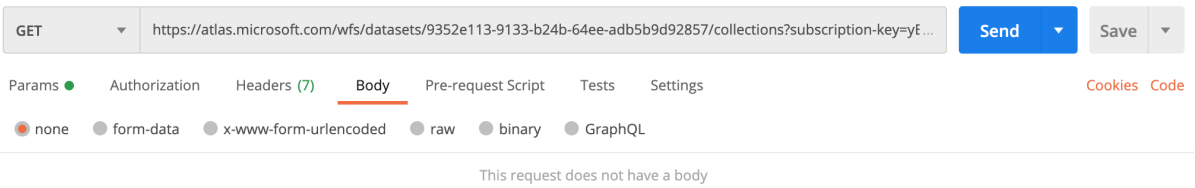
- 6.9 Query datasets with WFS API
    - 6.9.1 Make a GET request to view a list of the collections in the dataset
        - HTTP Method
            - GET
        - Request
            - https://atlas.microsoft.com/wfs/datasets/DATASETID/collections?subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY&api-version=1.0
            - Comment
                - 1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`

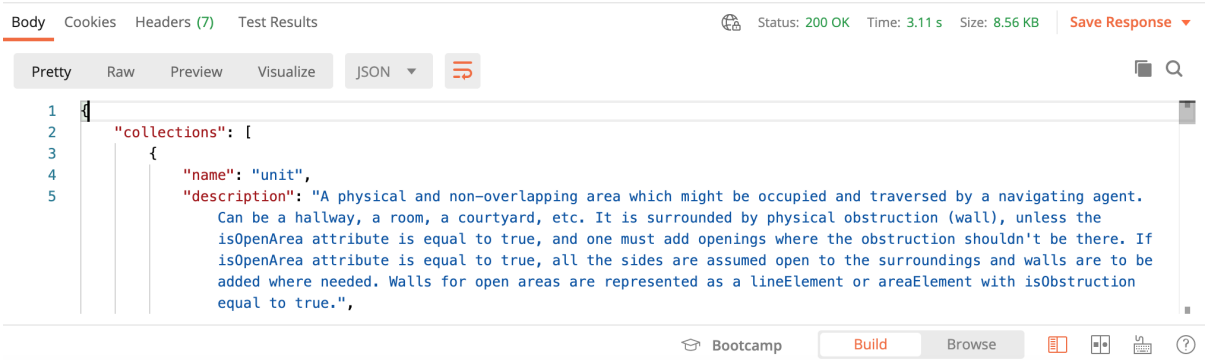


    - 2. Modify `DATASETID` to `DATASETID` that was retrieved in `Step 6.7.4`
  - 6.9.2 Click the blue `Send` button and wait for the request to process

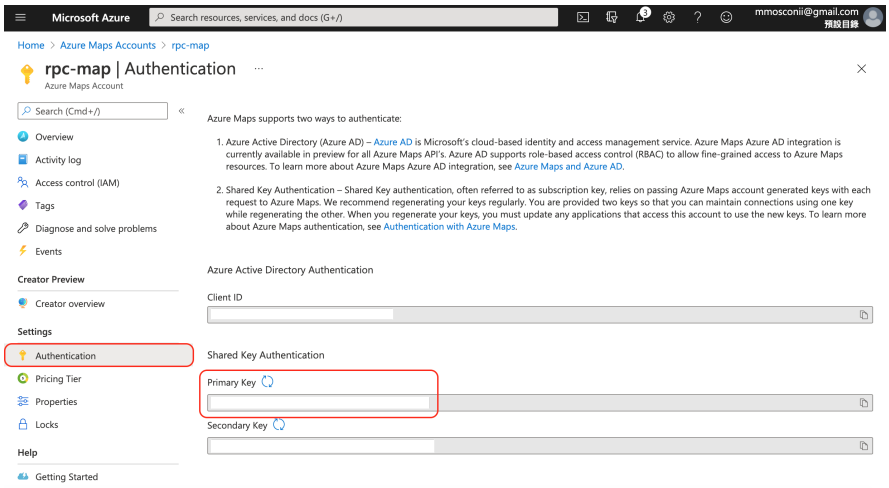


  - 6.9.3 Request complete

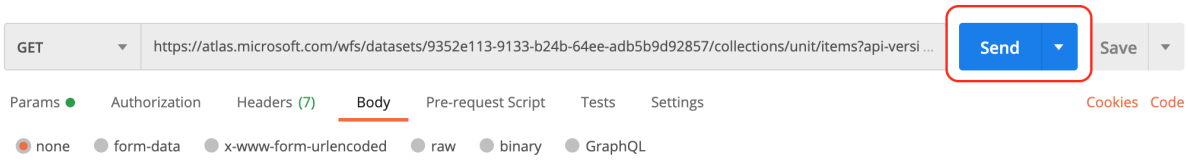




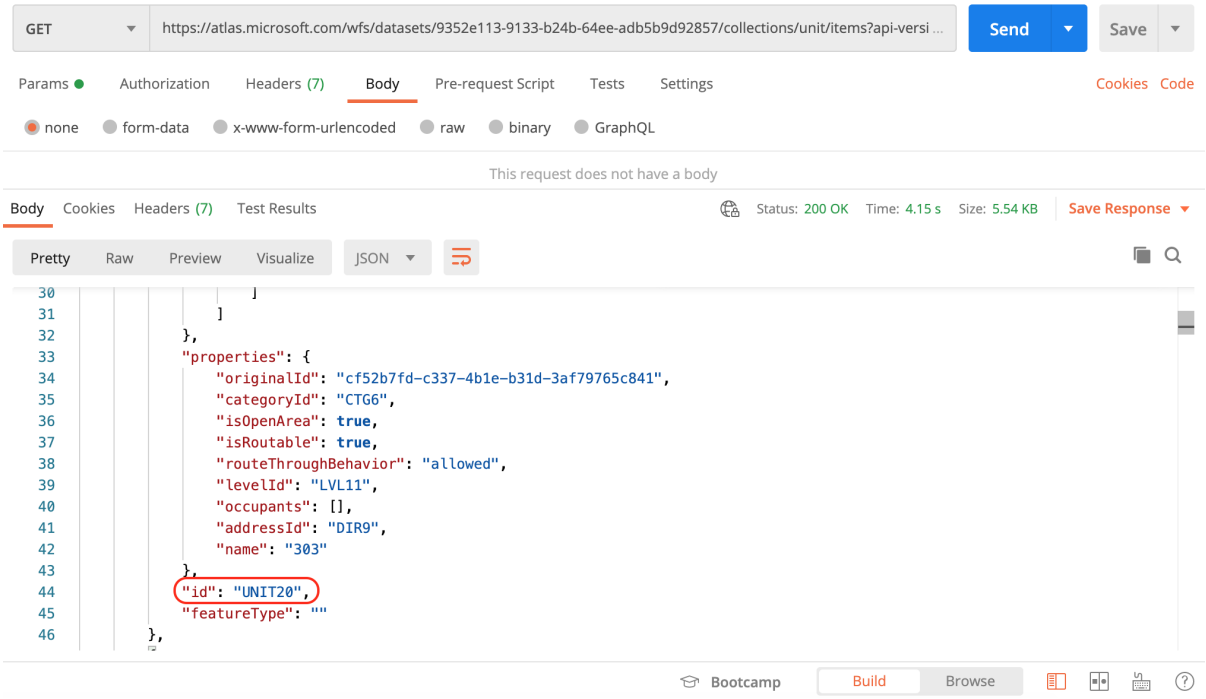
- 6.10 Make a request for the unit feature collections
  - 6.10.1 Make a GET request
    - HTTP Method
      - GET
    - Request
      - https://atlas.microsoft.com/wfs/datasets/DATASETID/collections/unit/items?subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY&api-version=1.0
      - Comment
        - 1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



    - 2. Modify `DATASETID` to `DATASETID` that was retrieved in `Step 6.7.4`
- 6.10.2 Click the blue `Send` button and wait for the request to process

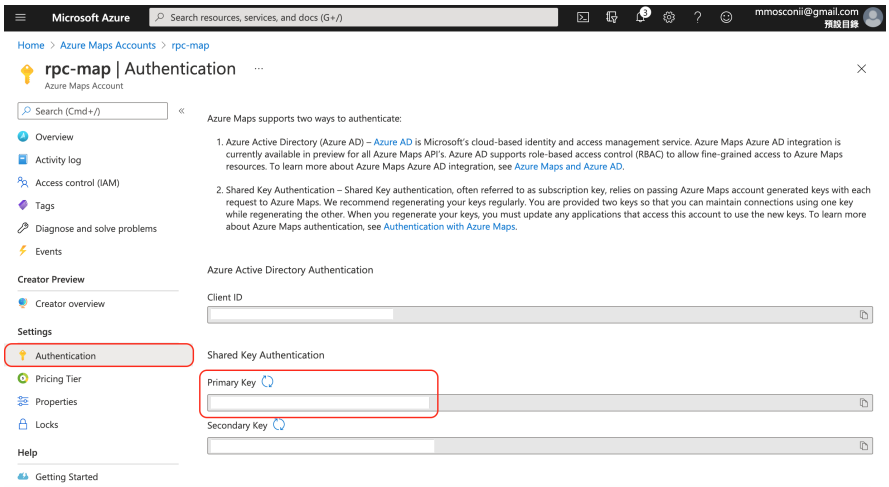


- 6.10.3 Request complete



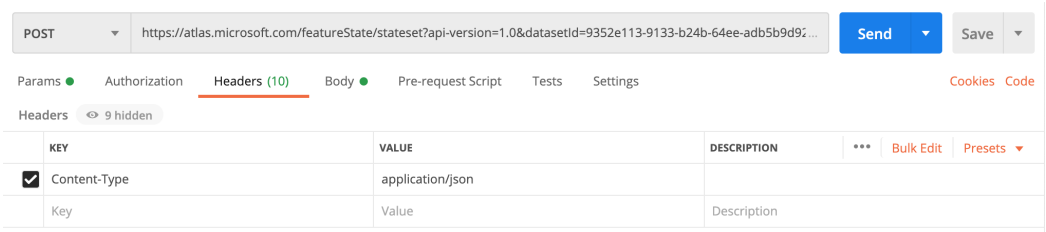
    - Comment
        - Copy the feature `id` for a unit feature, we'll use this feature id in the next section.
- 6.11 Create a feature stateset
    - 6.11.1 Make a POST request to the Create Stateset API
        - HTTP Method
            - POST
        - Request
            - https://atlas.microsoft.com/featureState/stateset?api-version=1.0&datasetId=DATASETID&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
            - Comment

        1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`



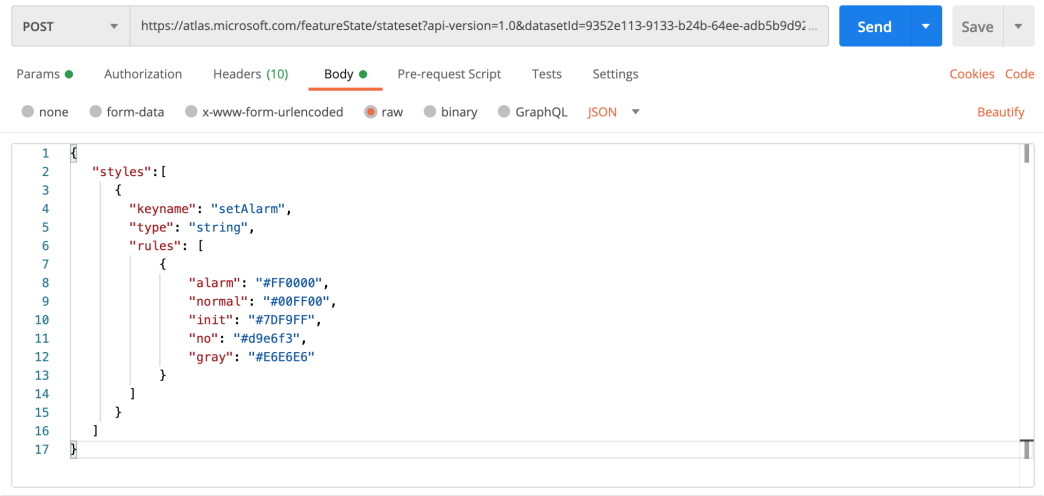
        2. Modify `DATASETID` to `DATASETID` that was retrieved in `Step 6.7.4`

- Headers
    - Content-Type: `application/json`

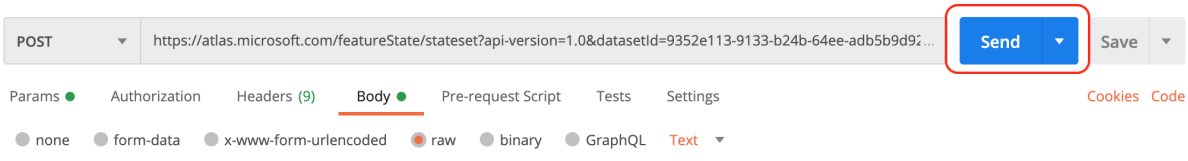


- Body
    - raw:

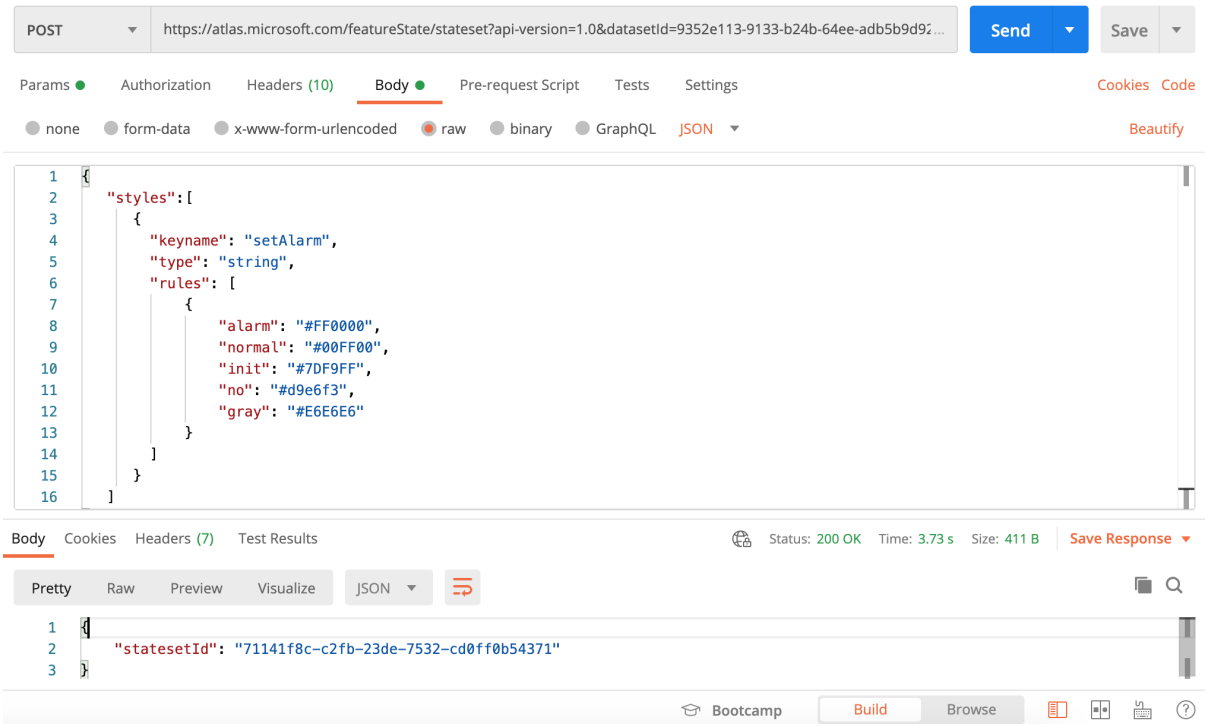
```
{
  "styles":[
      {
        "keyname": "setAlarm",
        "type": "string",
        "rules": [
            {
              "alarm": "#FF0000",
              "normal": "#00FF00",
              "init": "#7DF9FF",
              "no": "#d9e6f3",
              "gray": "#E6E6E6"
            }
        ]
      }
  ]
}
```



- 6.11.2 Click the blue Send button and wait for the request to process

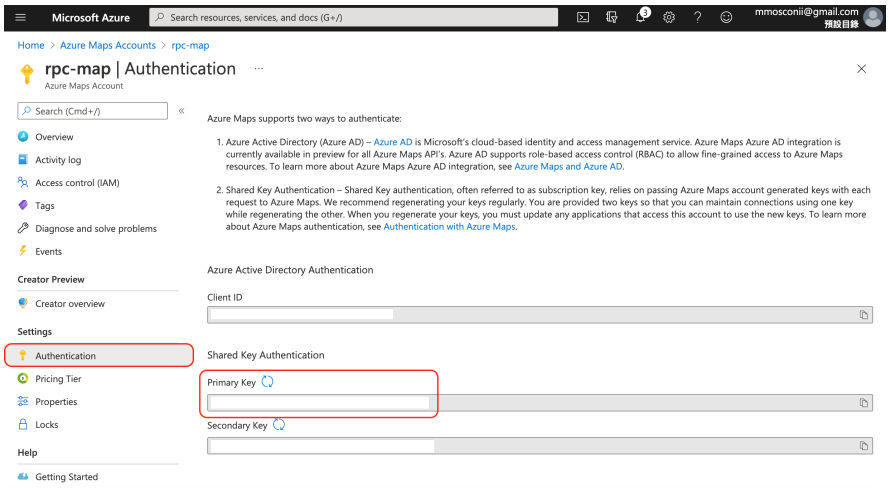


- 6.11.3 Request complete

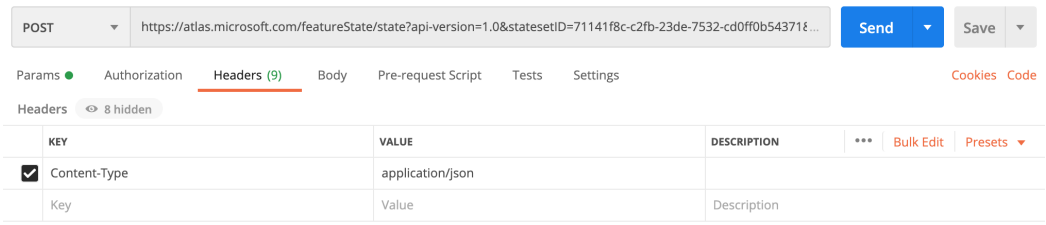


    - Comment
        - Copy the `statesetId` from the response body
- 6.12 Update the state
    - 6.12.1 Make a POST request to the Create Stateset API
        - HTTP Method
            - POST
        - Request
            - https://atlas.microsoft.com/featureState/state?api-version=1.0&statesetID=STATESETID&featureID=FEATUREID&subscription-key=AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY
            - Comment

        1. Modify `AZURE-MAPS-PRIMARY-SUBSCRIPTION-KEY` to `Primary Key`

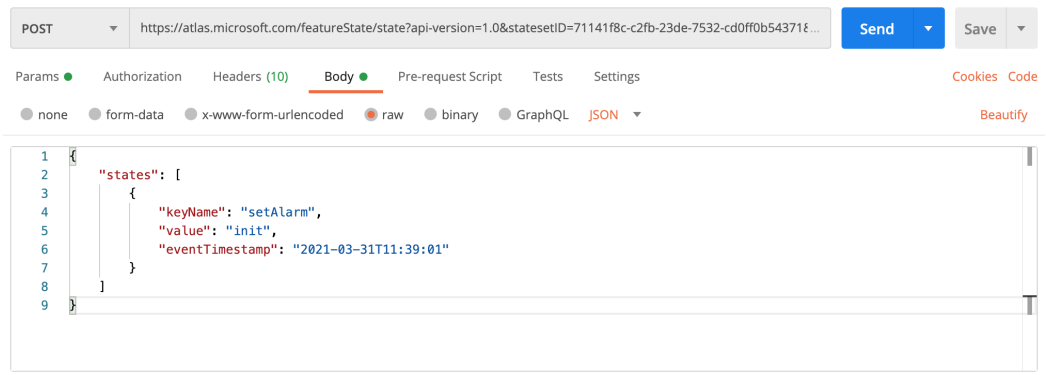


        2. Modify `STATESETID` to `statesetId` that was retrieved in `Step 6.11.3`
        3. Modify `FEATUREID` to `id` that was retrieved in `Step 6.10.3`
- Headers
    - Content-Type: `application/json`

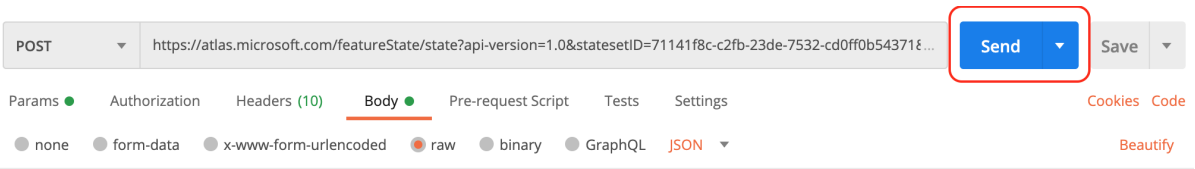


- Body
    - raw

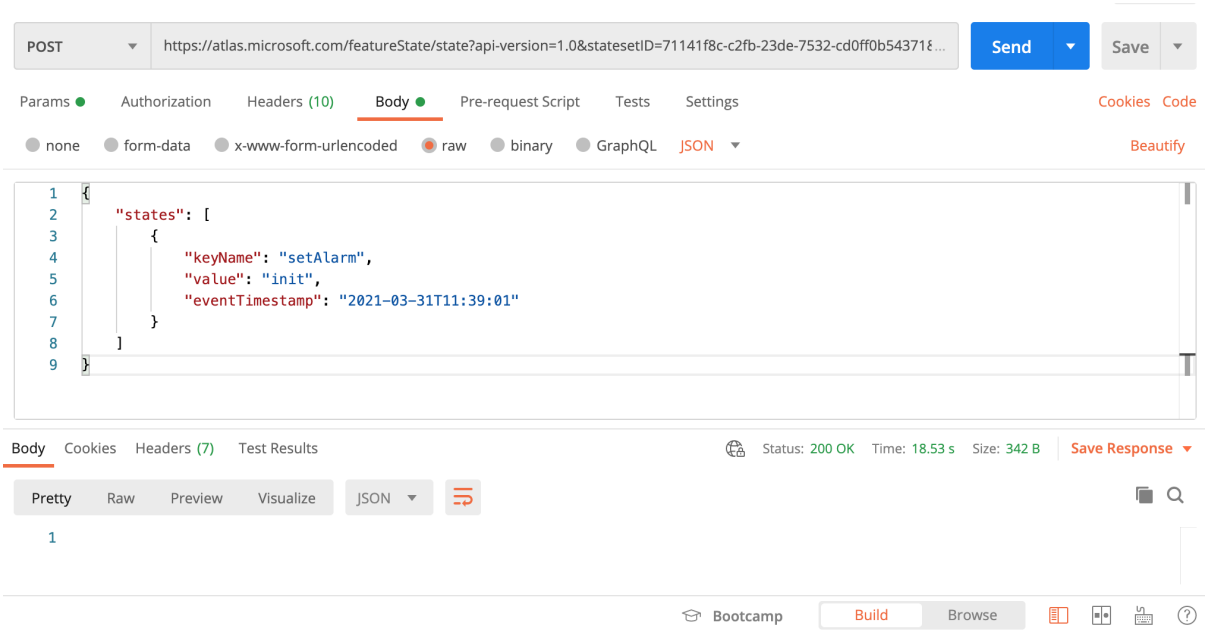
```
{
  "states": [
    {
        "keyName": "setAlarm",
        "value": "init",
        "eventTimestamp": "2021-03-31T11:39:01"
    }
  ]
}
```



        - Comment
            - The update will only be saved if the time posted stamp is after the time stamp of the previous request.
    - 6.12.2 Click the blue `Send` button and wait for the request to process

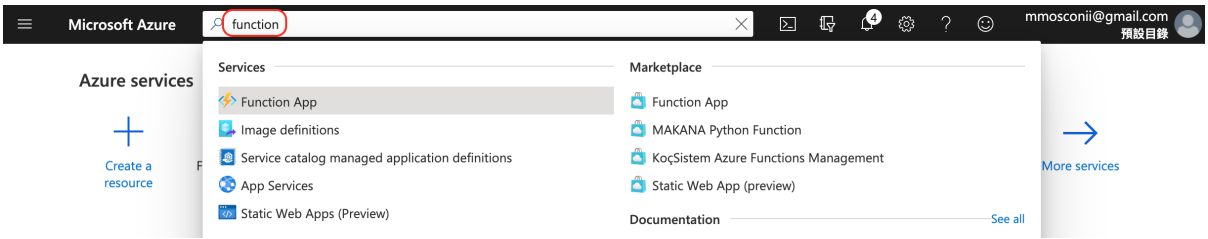


    - 6.12.3 Request complete

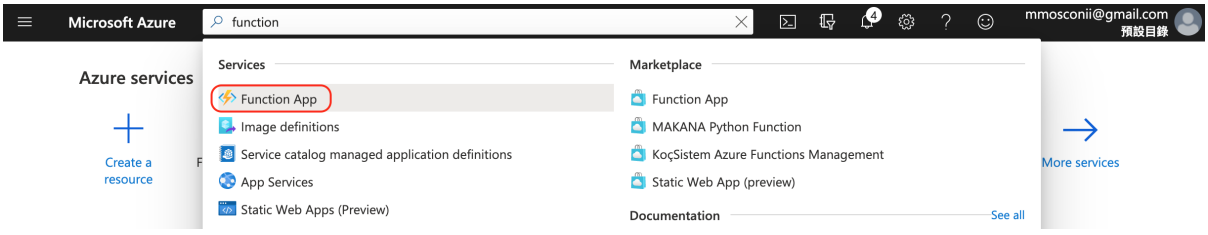


7. Create & Deploy toADT Function

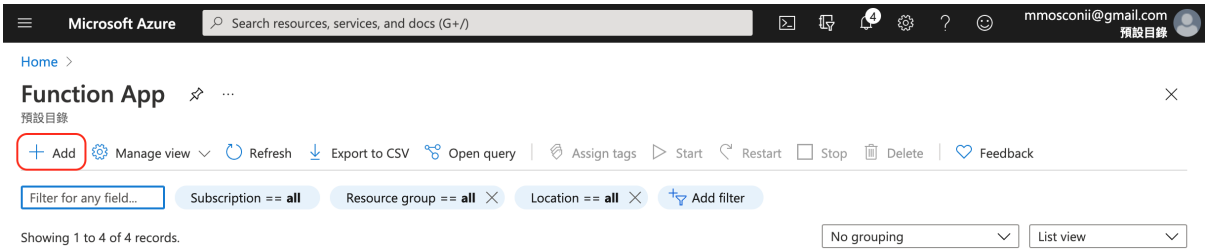
- 7.1 Create Azure Functions
    - Search `function`



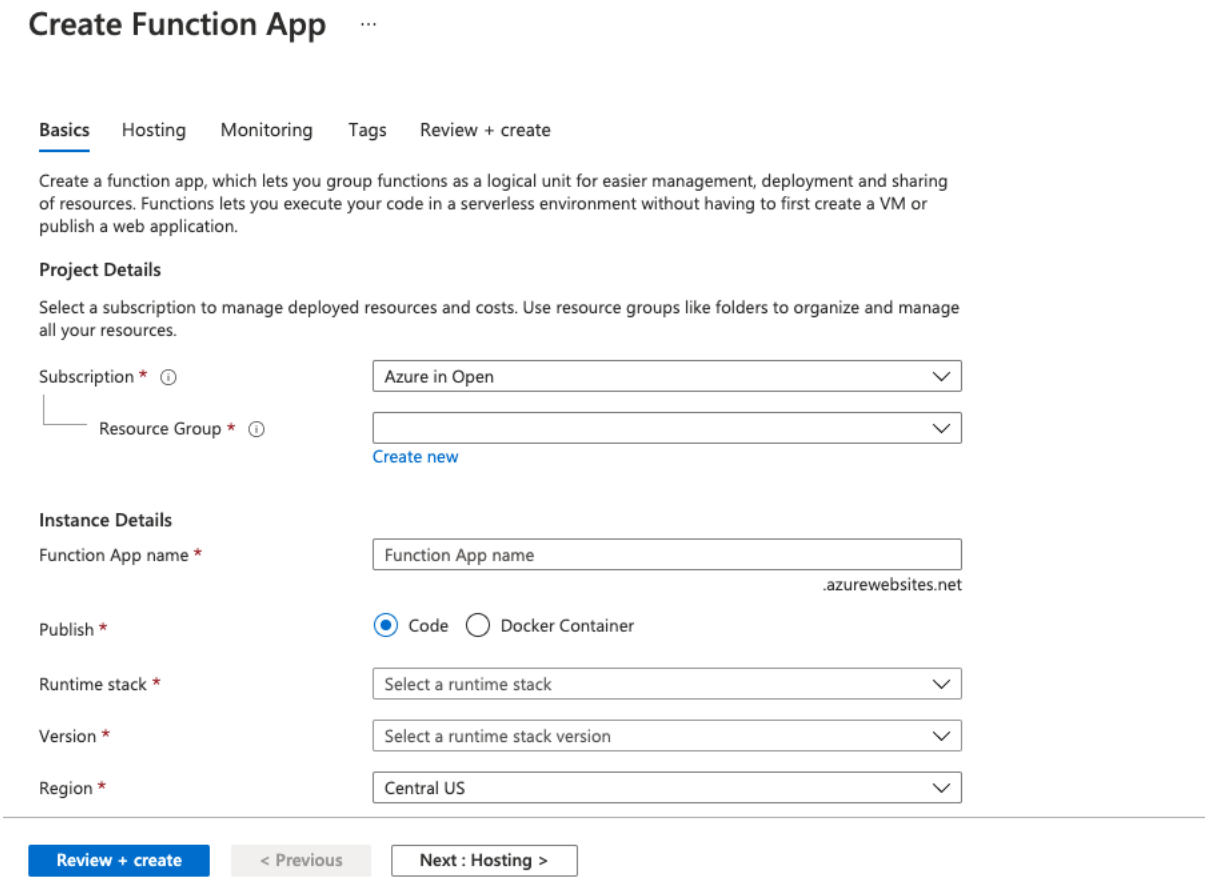
    - Select `Function App`



- Click `Add`

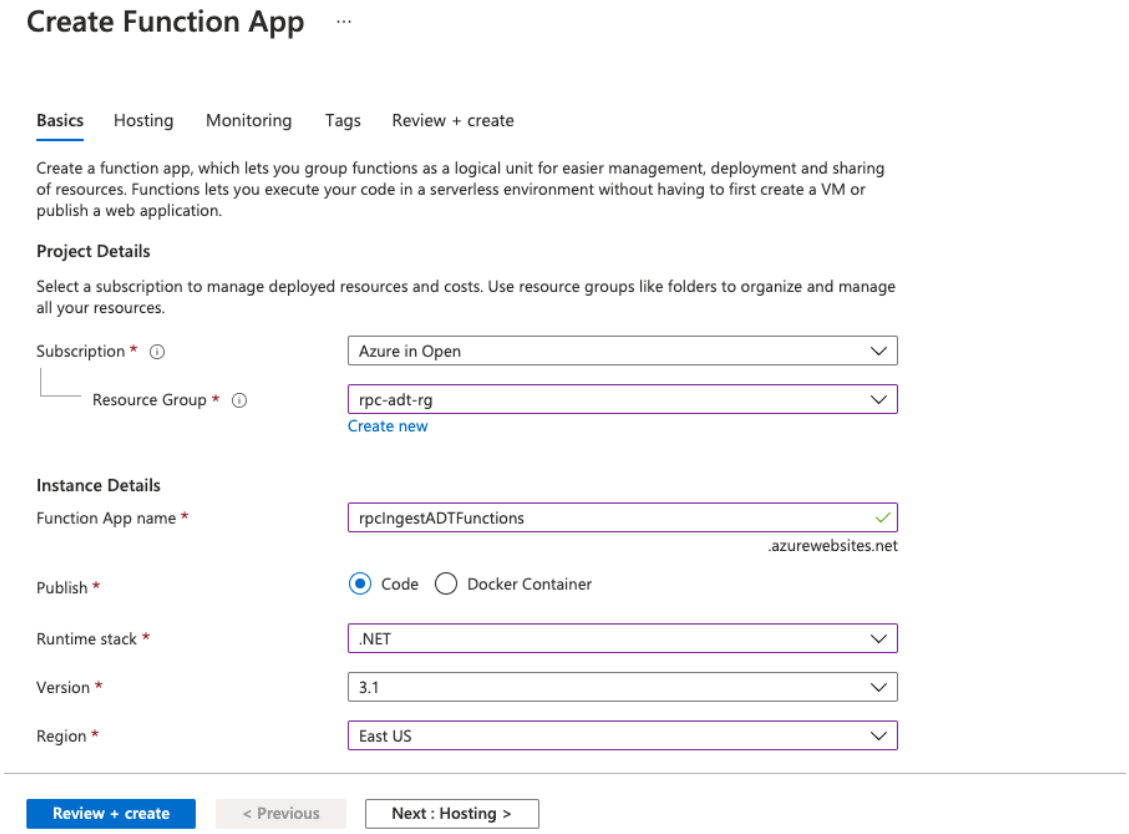


- Input data

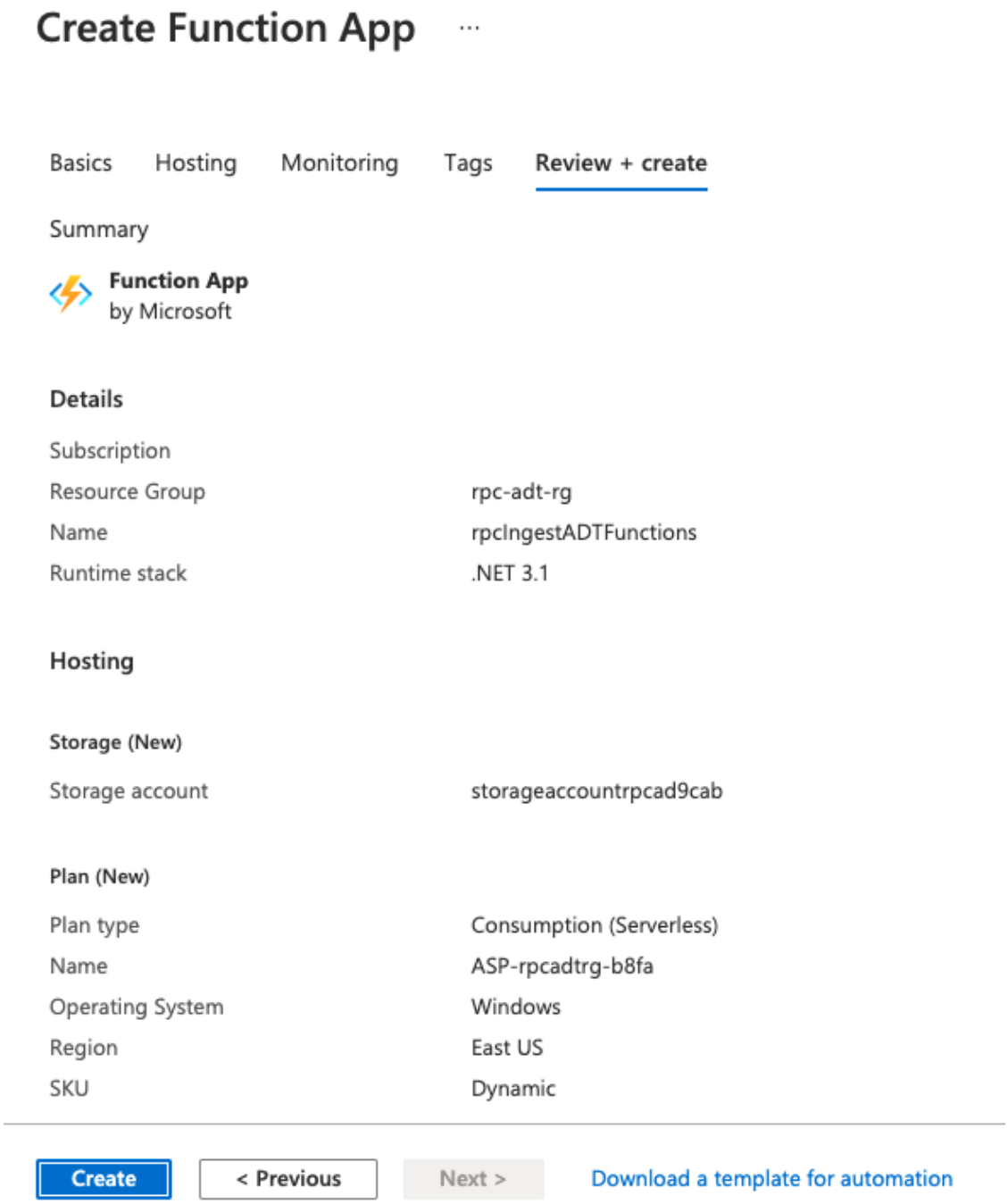


  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name, this example uses `rpc-adt-rg`
  - `Function App name` field
    - Please use a recognizable name, this example uses `rpcIngestADTFunctions`
  - `Publish` field
    - Select `Code`
  - `Runtime stack` field
    - Select `.NET`
  - `Version` 欄位
    - Select `3.1`
  - `Region` field
    - This example uses `East US`

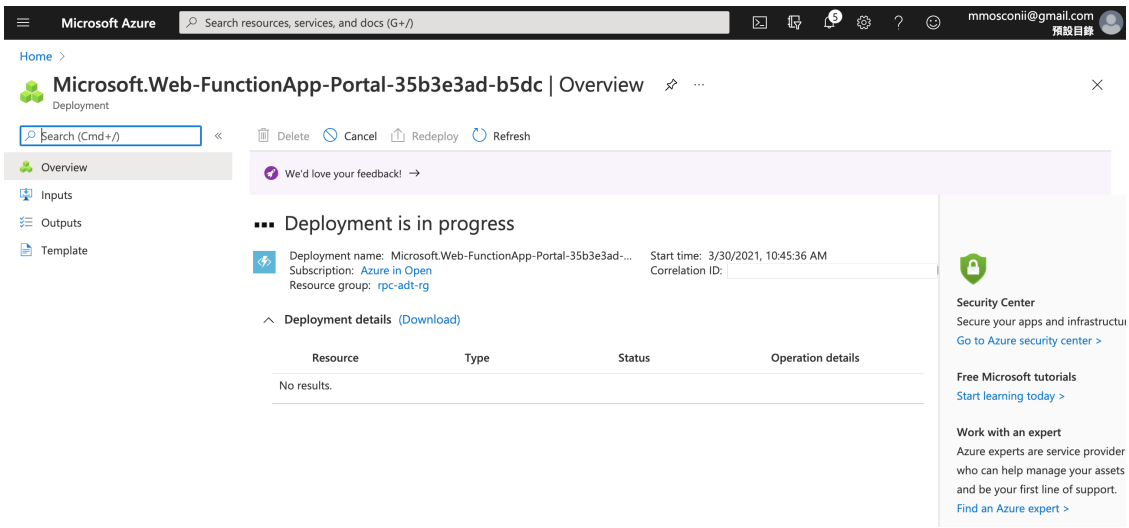
- When the input is complete, please click the `Review + create` button



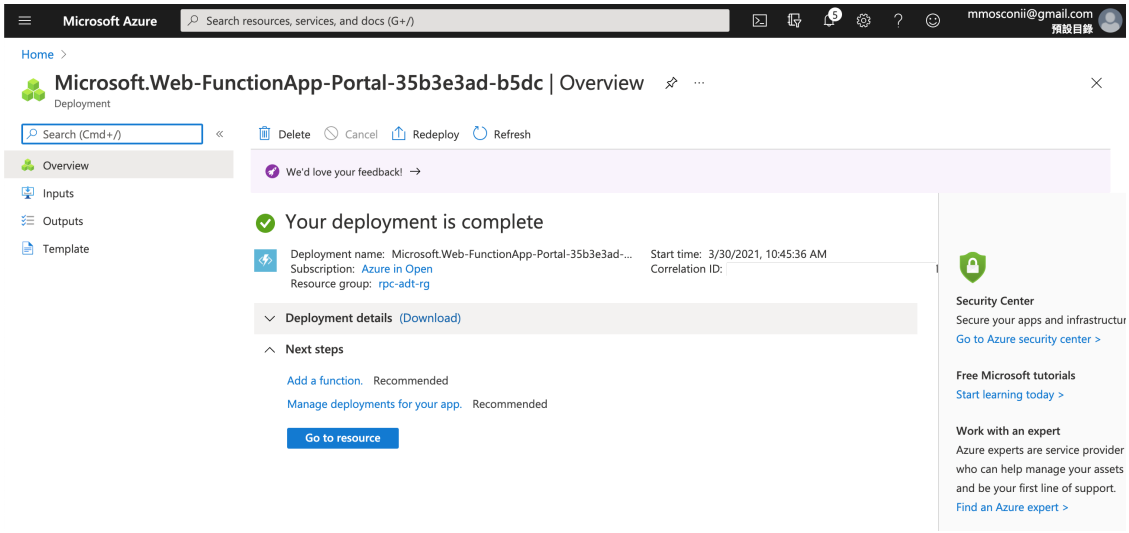
- Review your settings and select `Create`



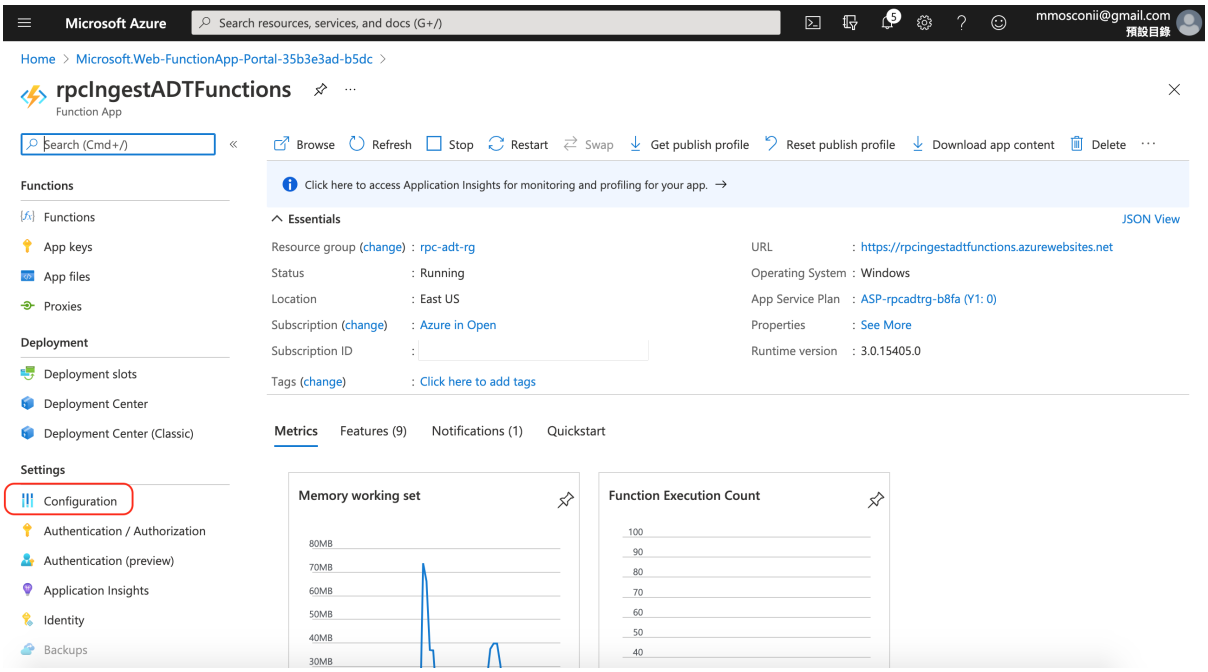
- Wait for the request to process



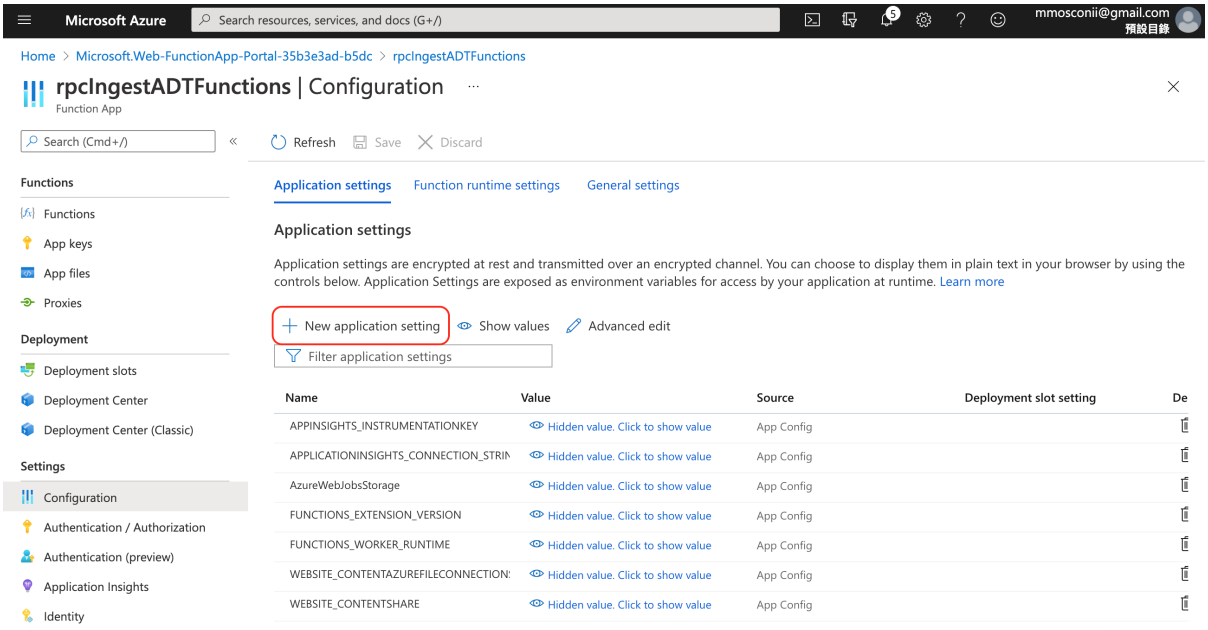
- Once deployment complete click on `Go to resource` button



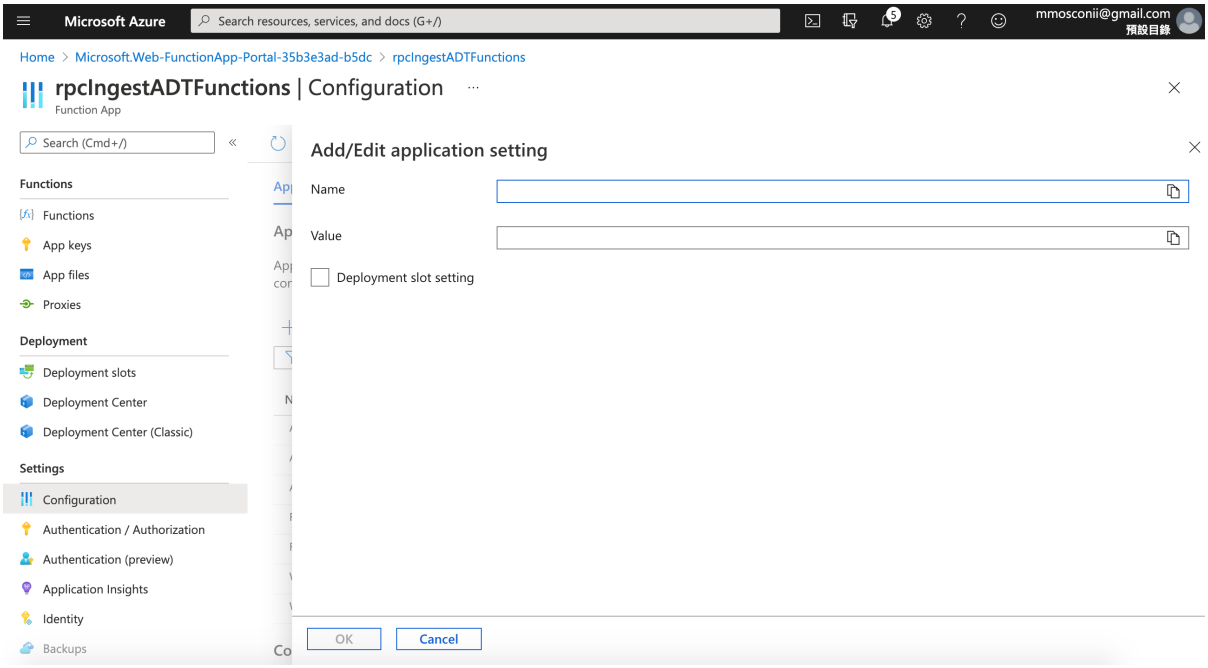
- 7.2 Set Configuration
  - Click `Configuration`



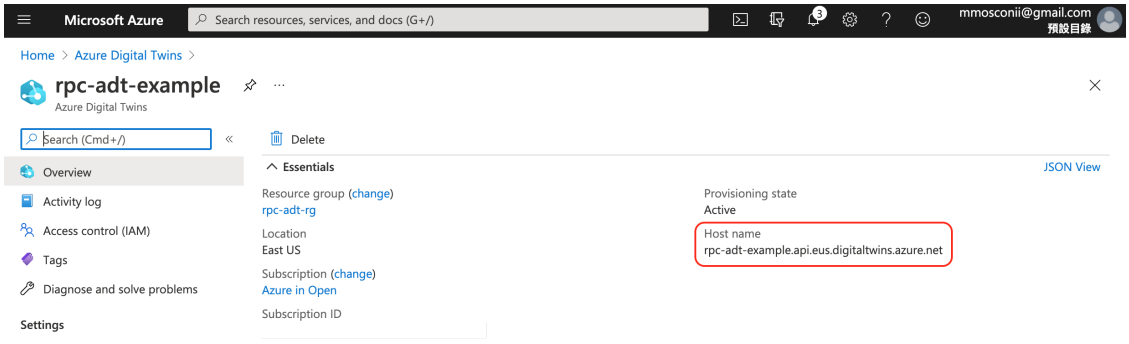
- Click `New application setting`



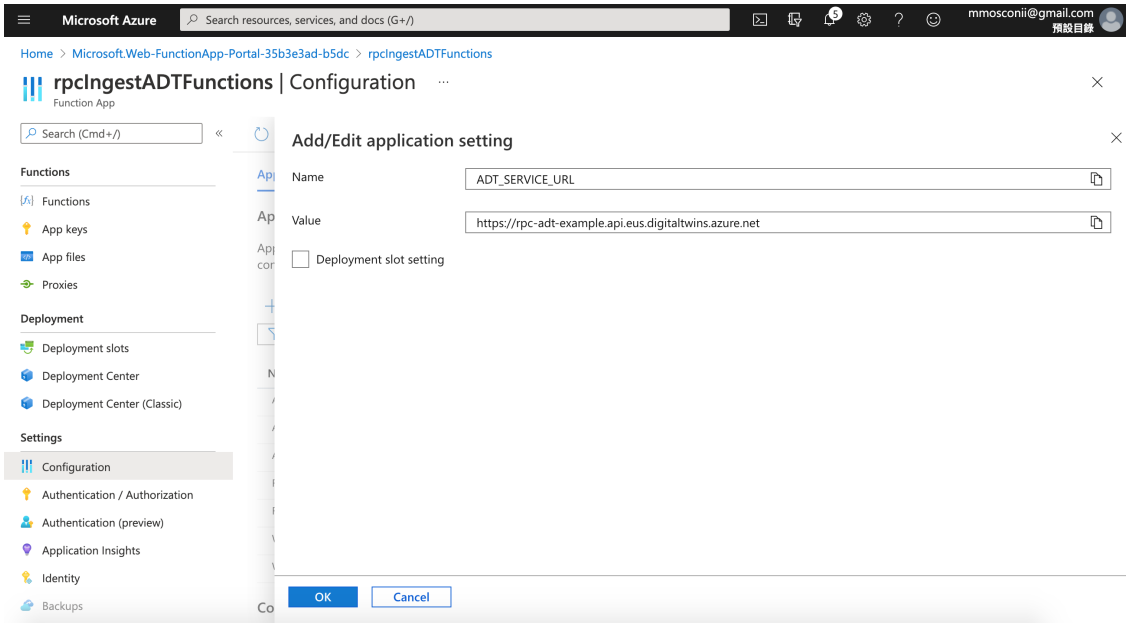
- Input data



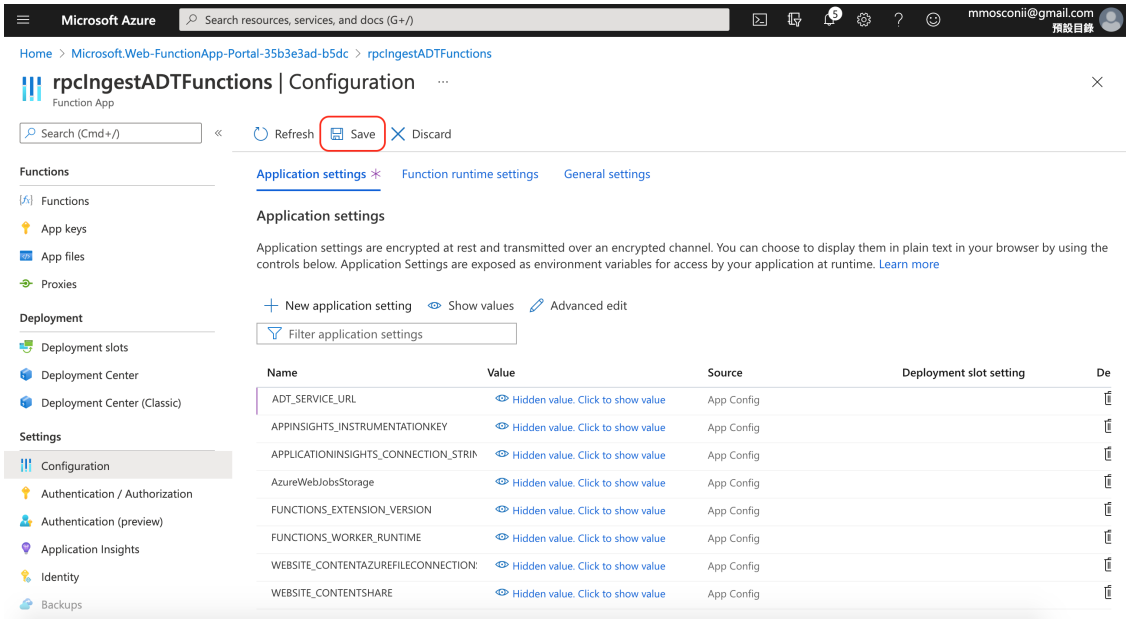
  - In the `Name` field, please enter `ADT_SERVICE_URL`
  - In the `Value` field (Add prefix `https://` in URL)



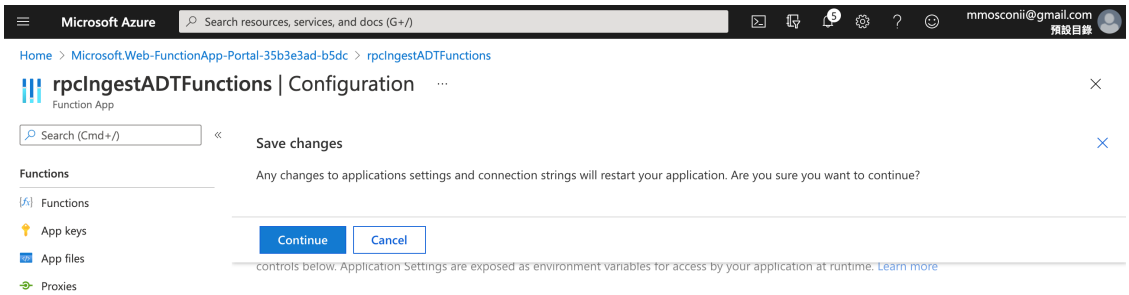
- Please click OK button after entering the above information



- Click Save

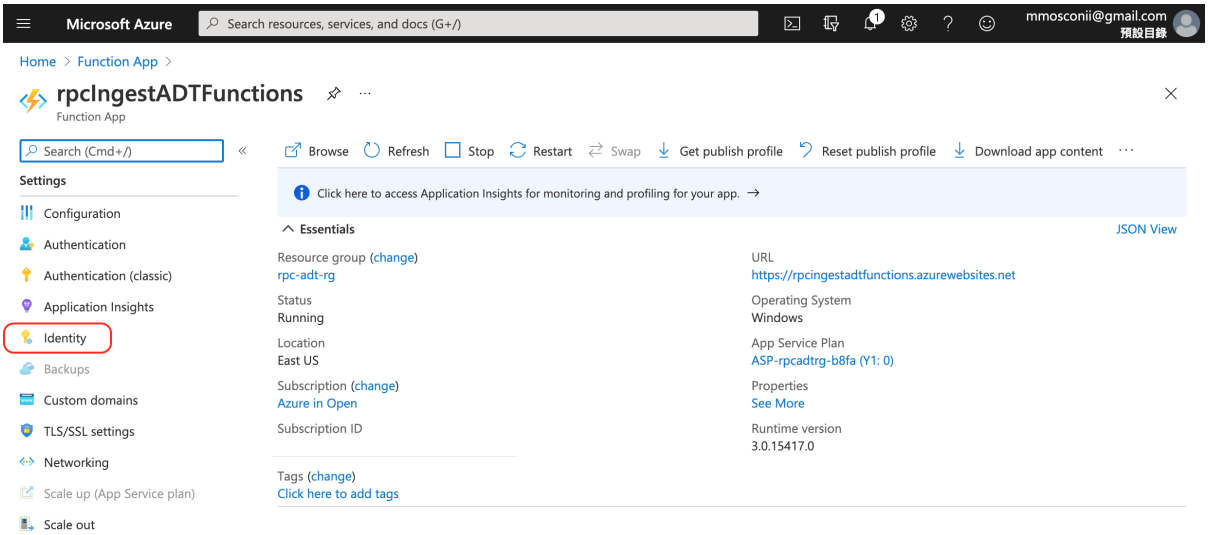


- Click Continue

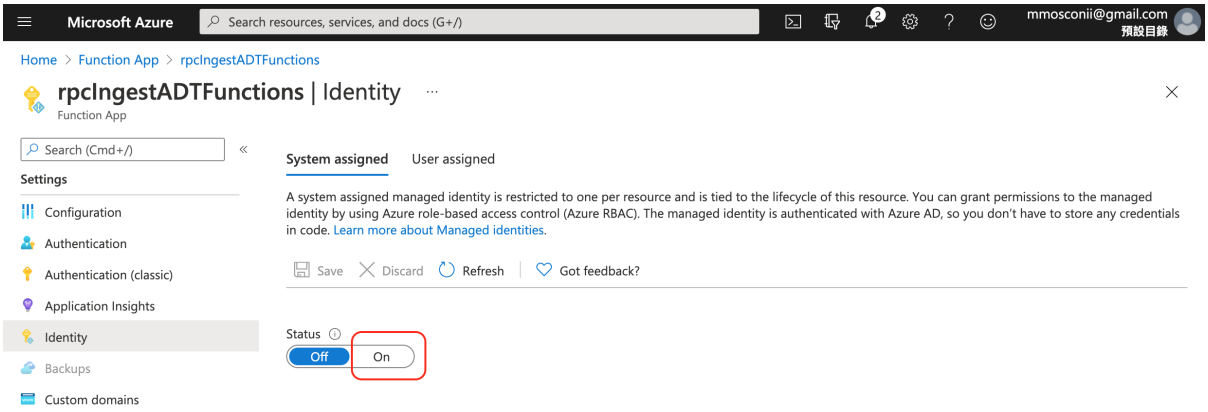


- 7.3 Assign access role

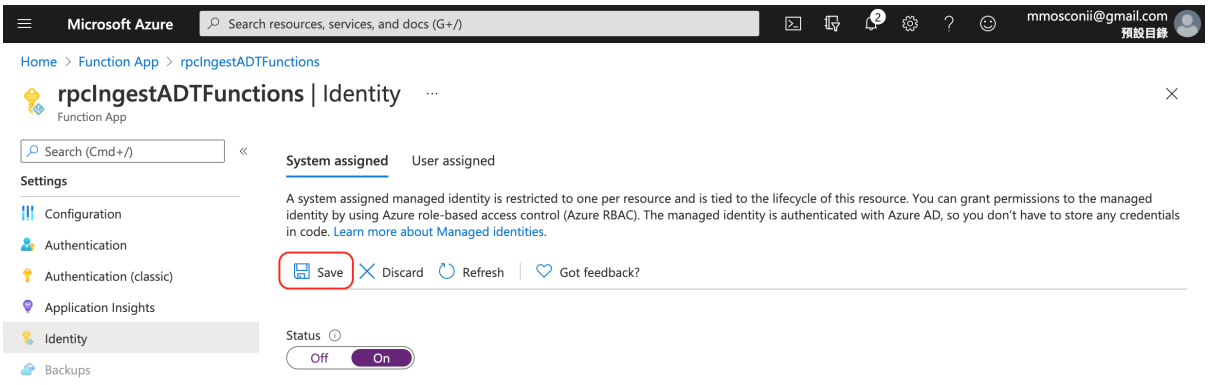
- Select `Identity` in the navigation bar on the left to work with a managed identity for the function



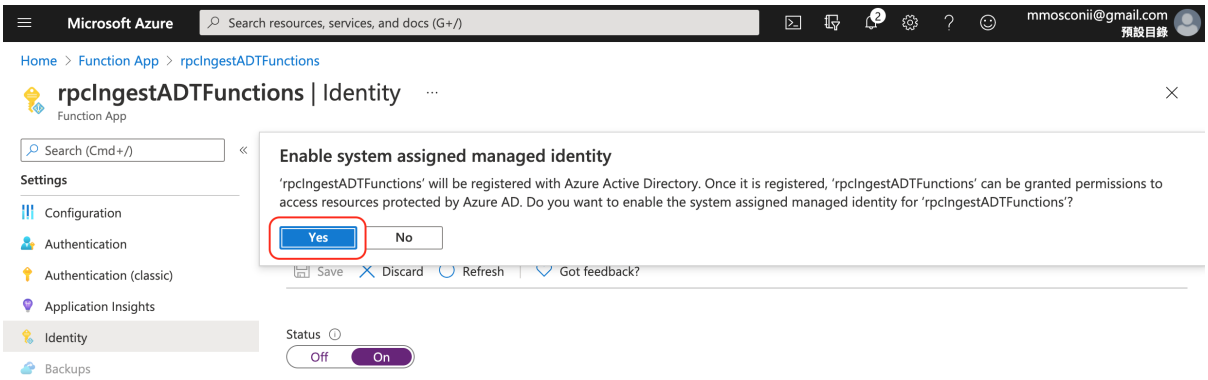
- Click the `On` button



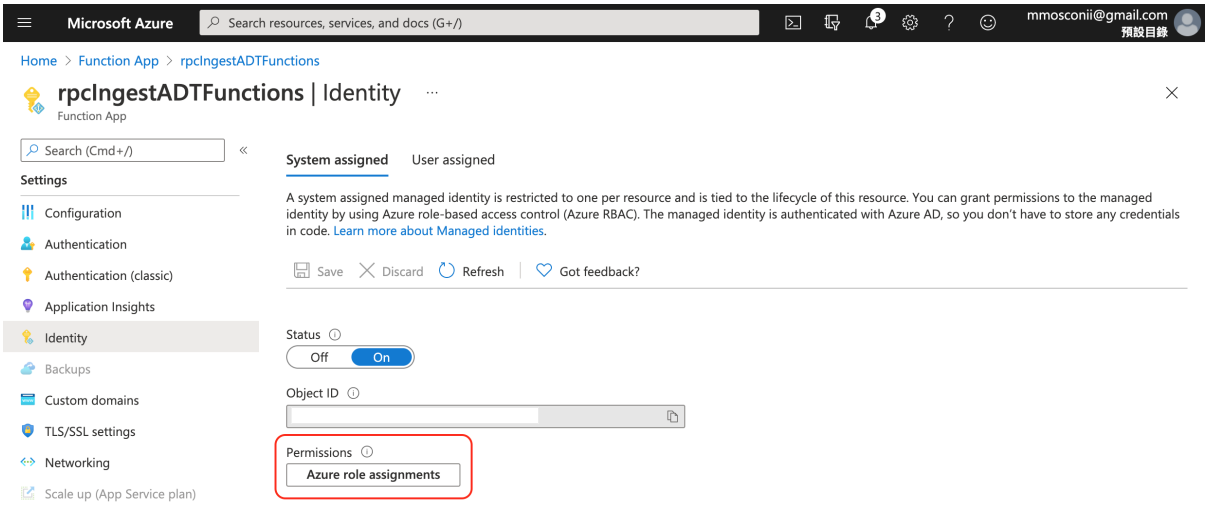
- Click the `Save` button



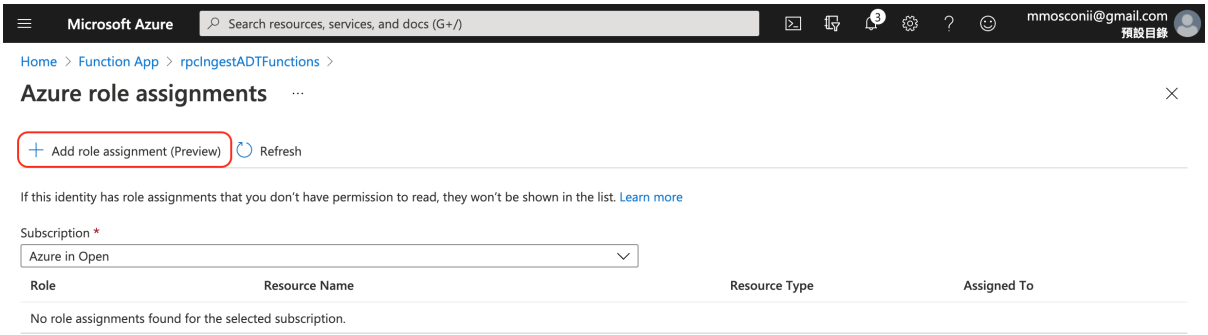
- Click the `Yes` button



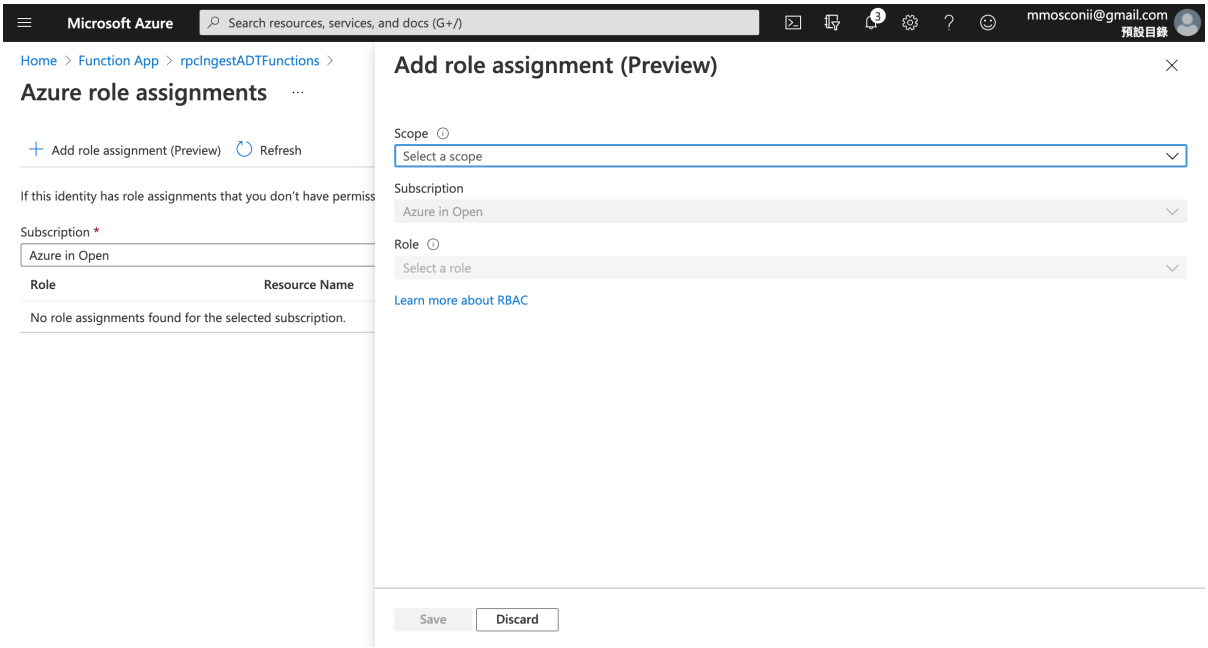
- Select the `Azure role assignments` button



- Select the `+ Add role assignment (Preview)` button

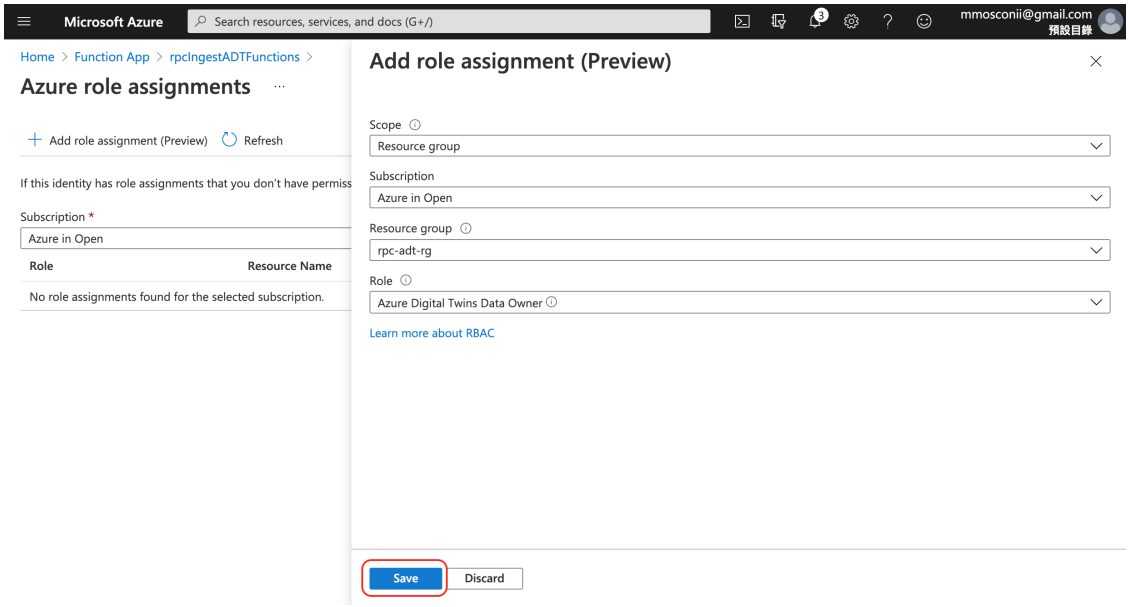


- Input data

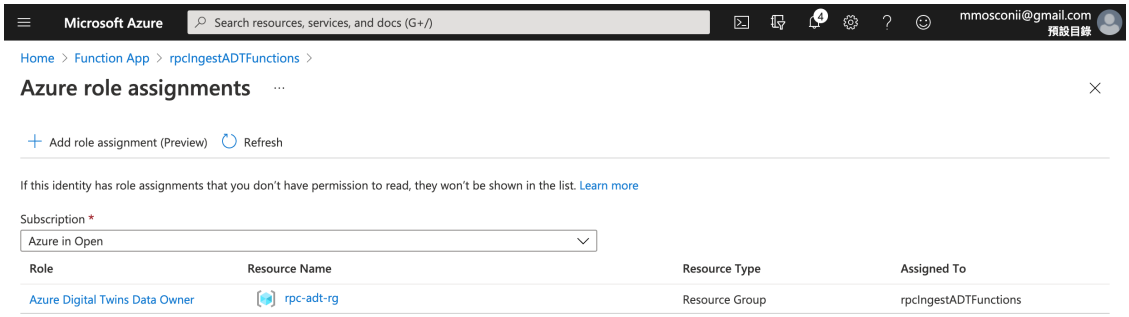


  - `Scope`: Select `Resource group`
  - `Subscription`: Select your Azure subscription
  - `Resource group`: Select your resource group from the dropdown
  - `Role`: Select `Azure Digital Twins Data Owner` from the dropdown

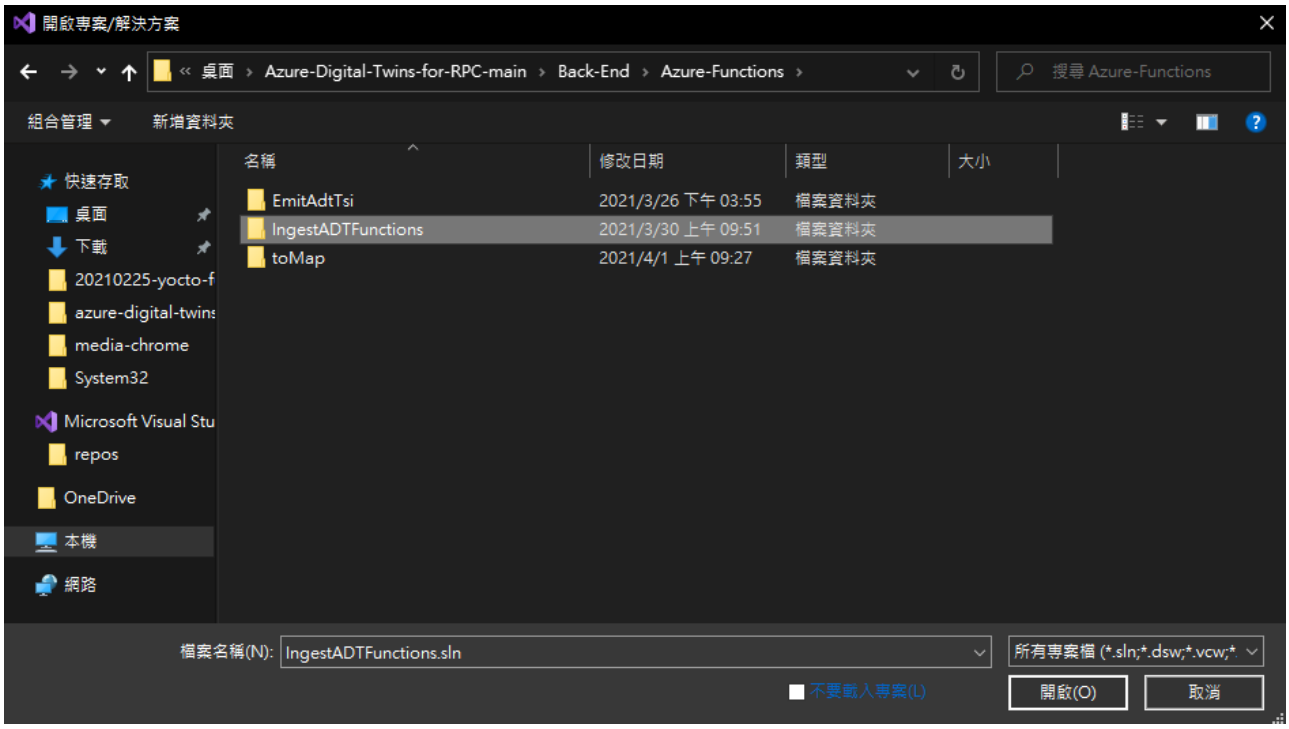
- When the input is complete, please click the `Save` button

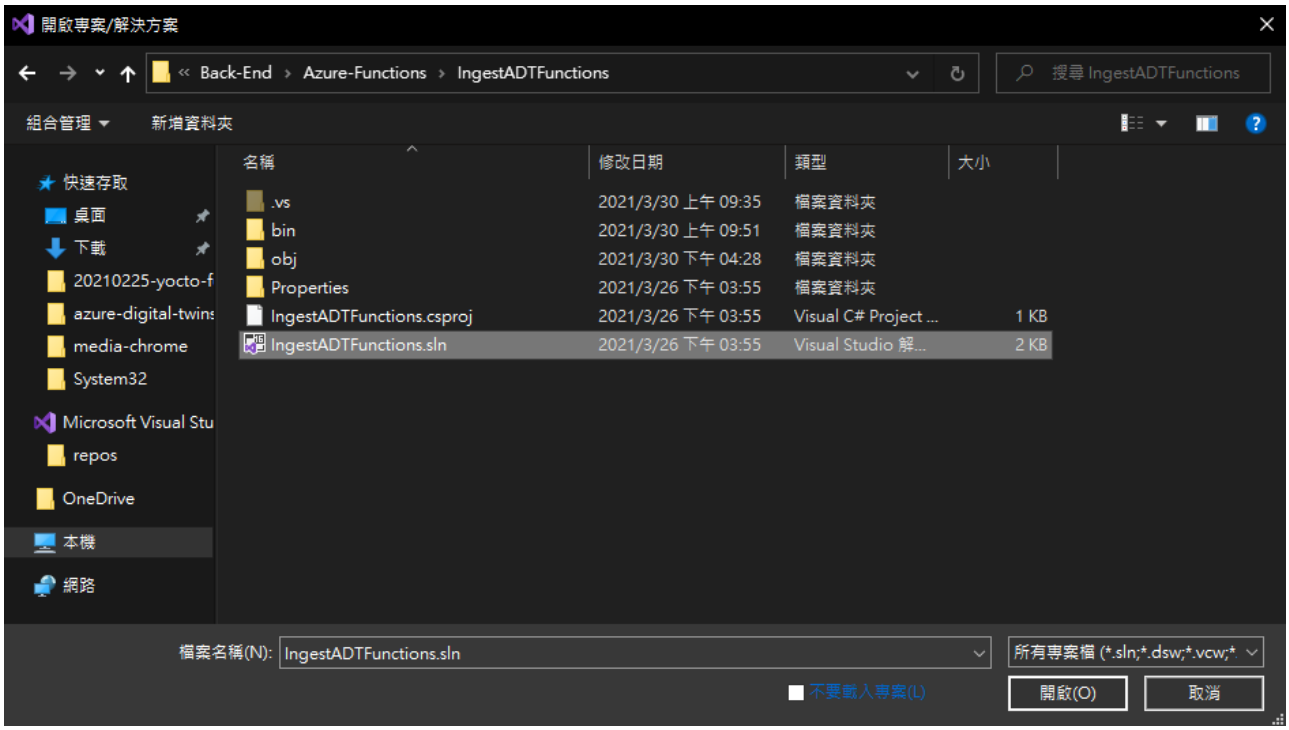


- Save finish



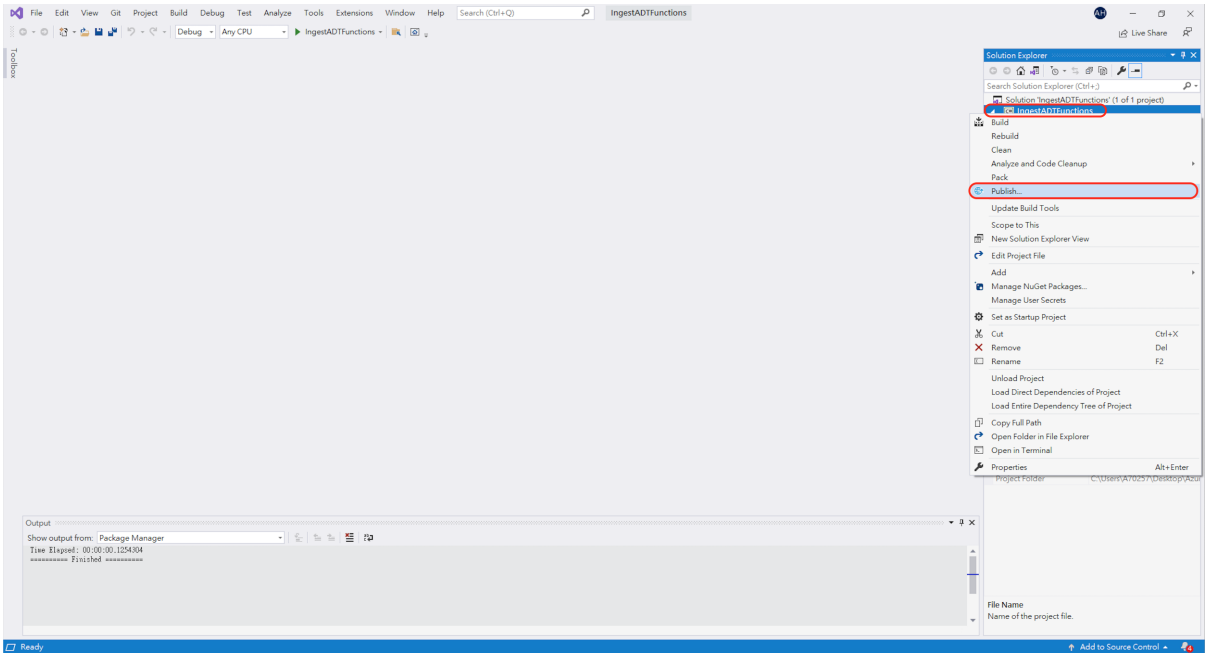
- 7.4 Opening `IngestADTFunctions.sln` from `/Back-End/Azure-Functions/IngestADTFunctions` in `Visual Studio 2019`



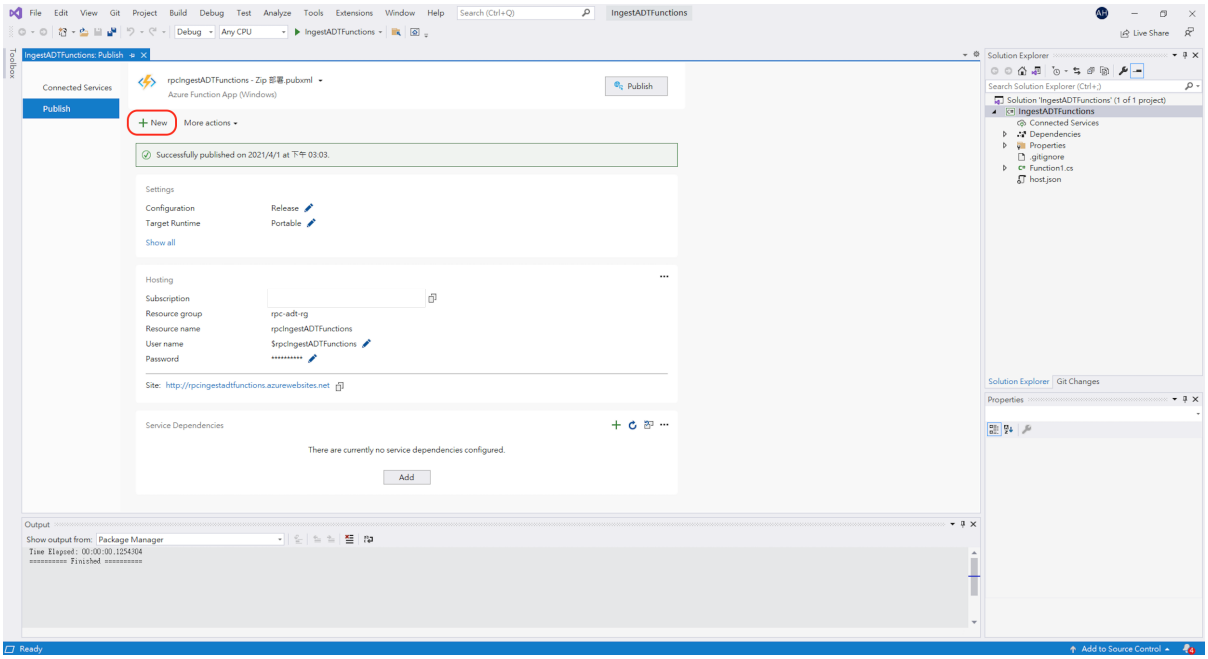


- 7.5 Deploy `IngestADTFunctions` to `Azure Function`

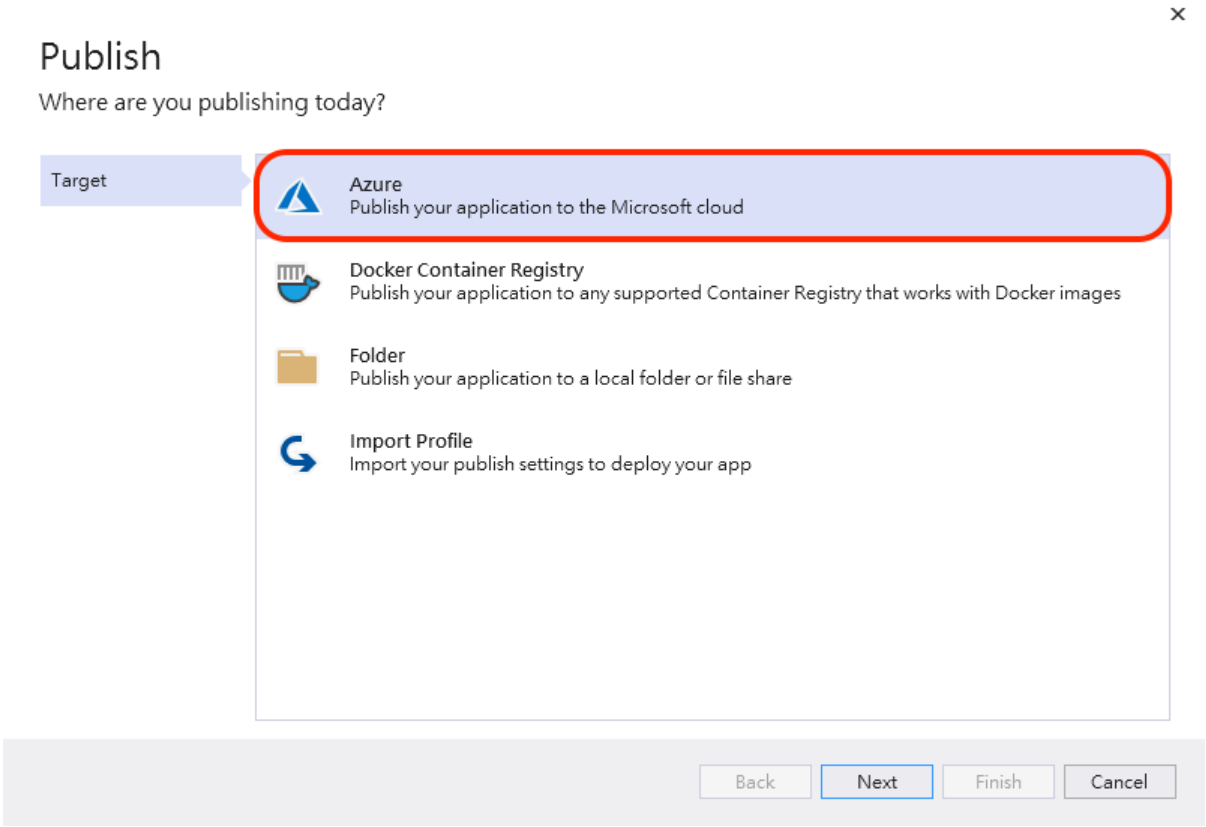
- Click `Publish...`



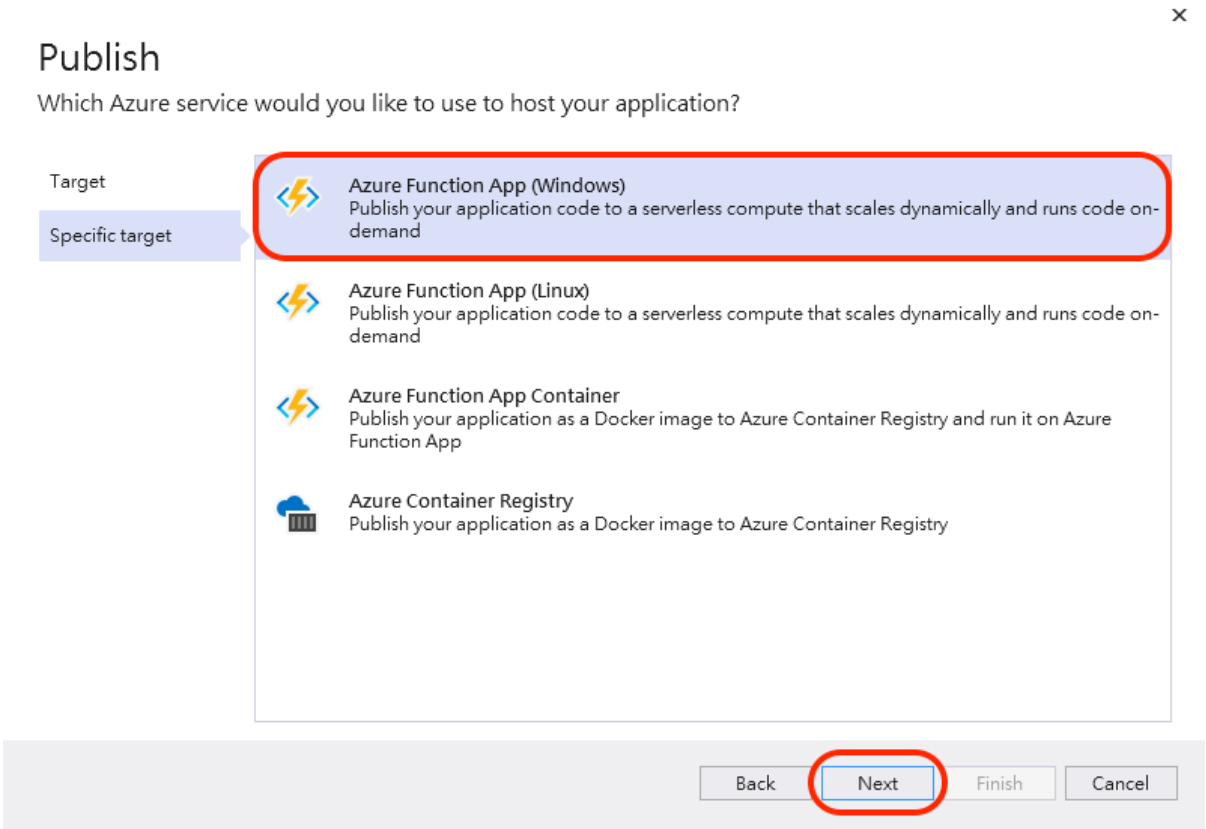
- Click `+ New`



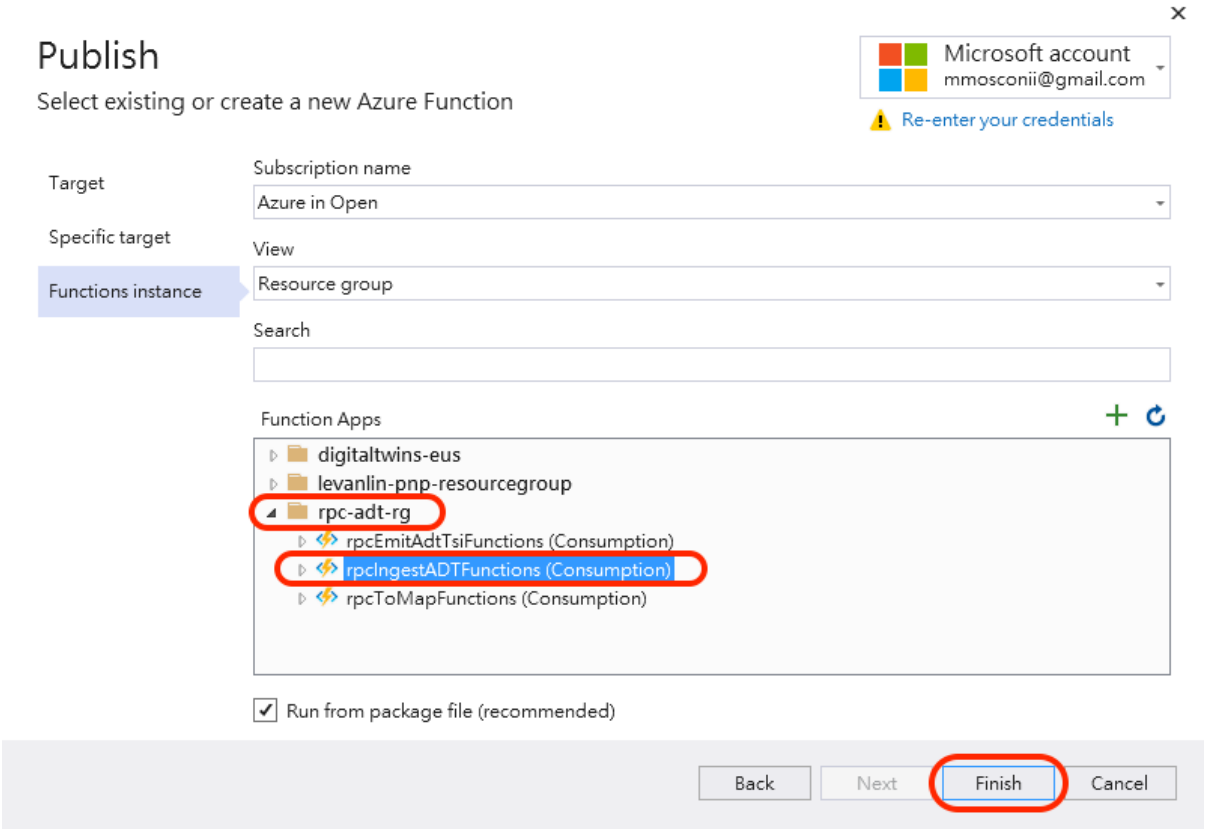
- Click `Azure`



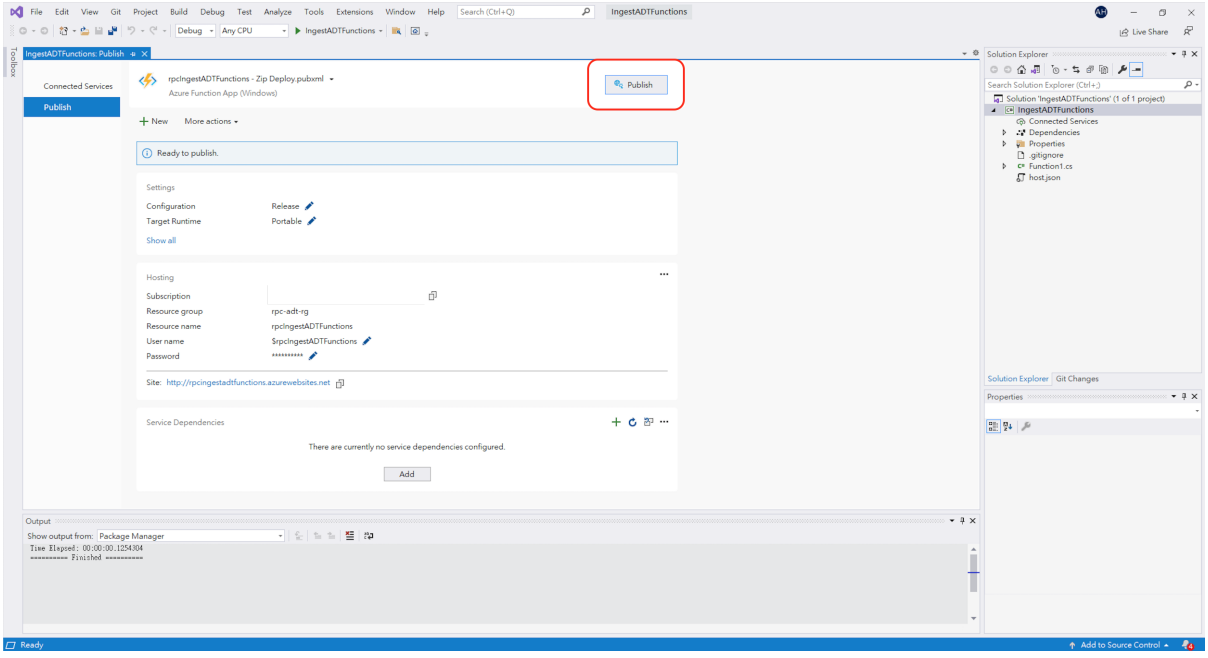
- Click `Azure Function App (Windows)` > `Next`



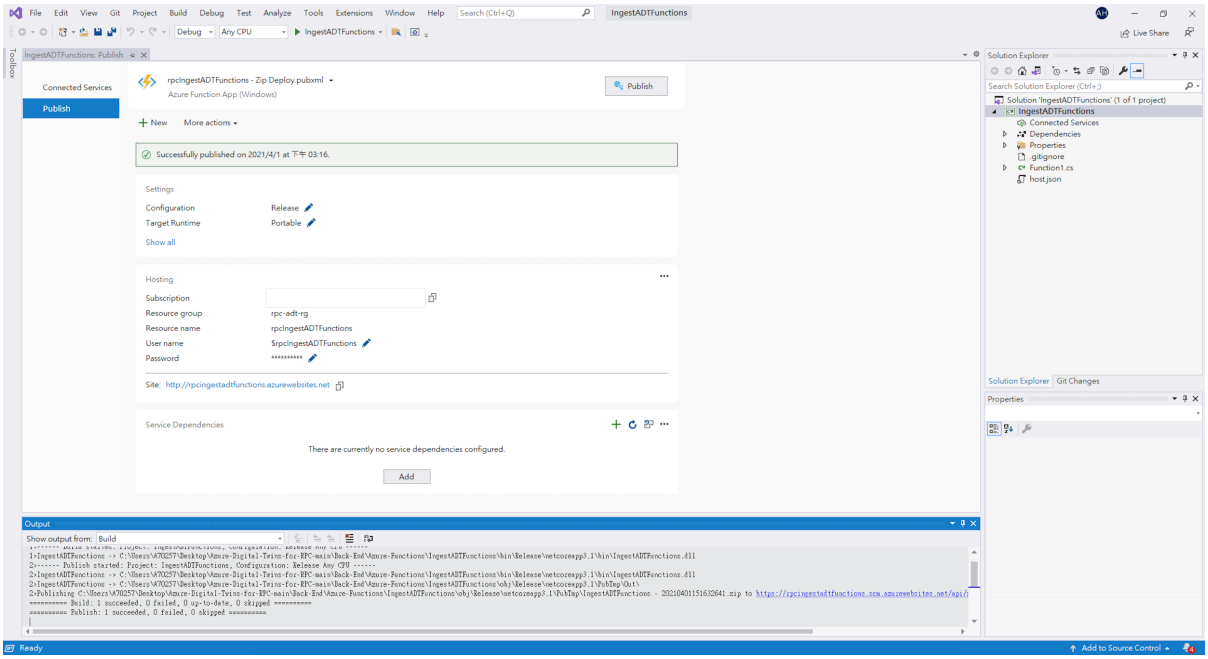
- Please select the function created in `Step 7.1`



- Click `Publish`

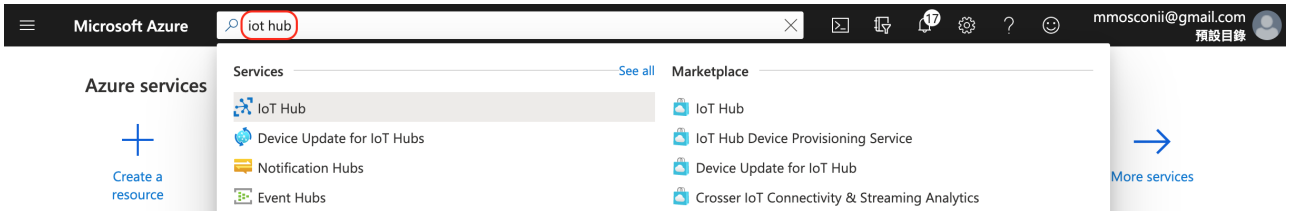


- Operation completes

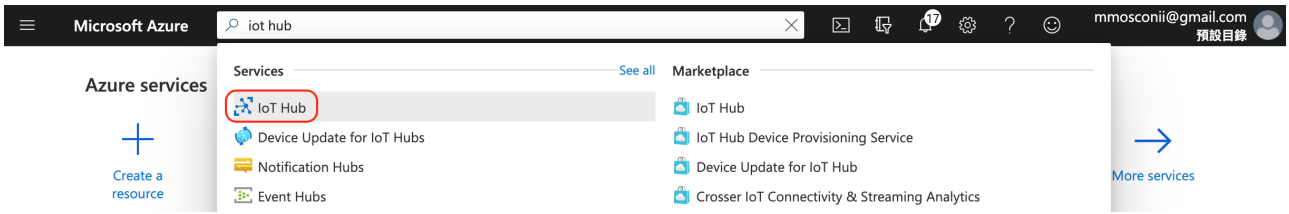


8. Create Azure IoT Hub, Set Event Grid, Create IoT Device

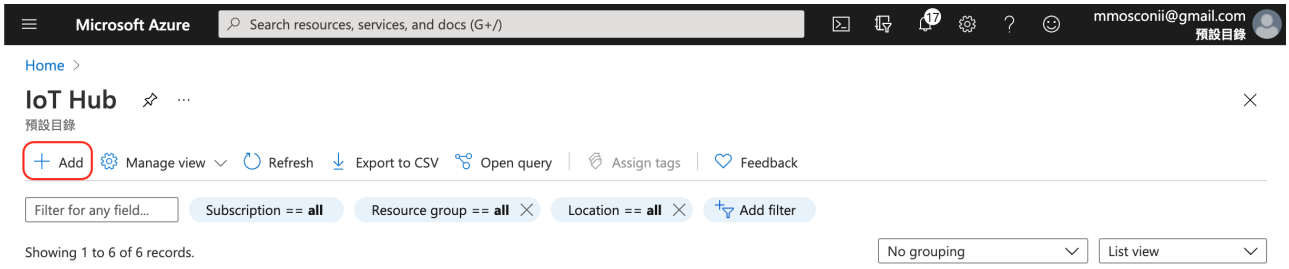
- 8.1 Search `iot hub`



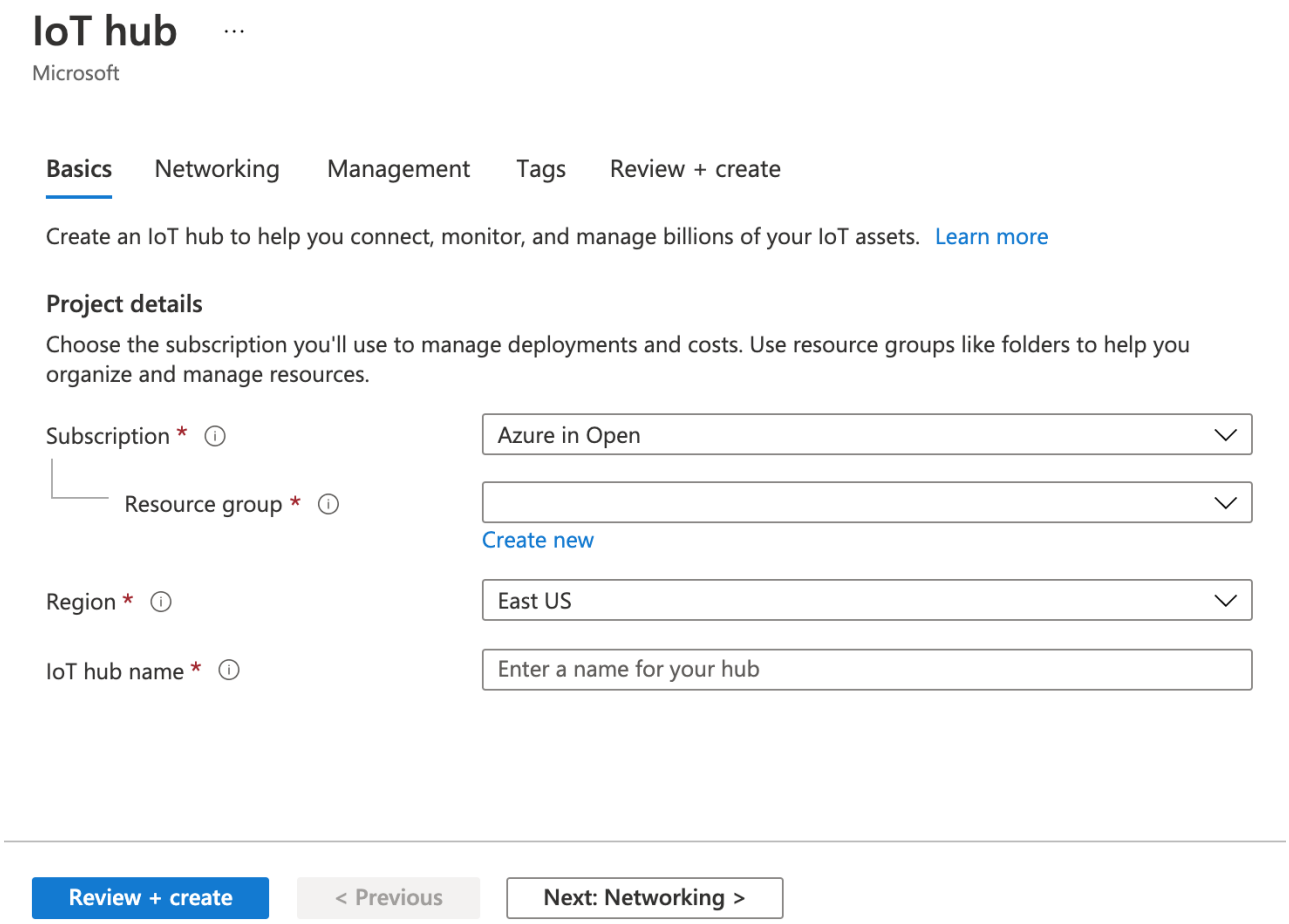
- 8.2 Select `IoT Hub`



- 8.3 Click `+ Add`

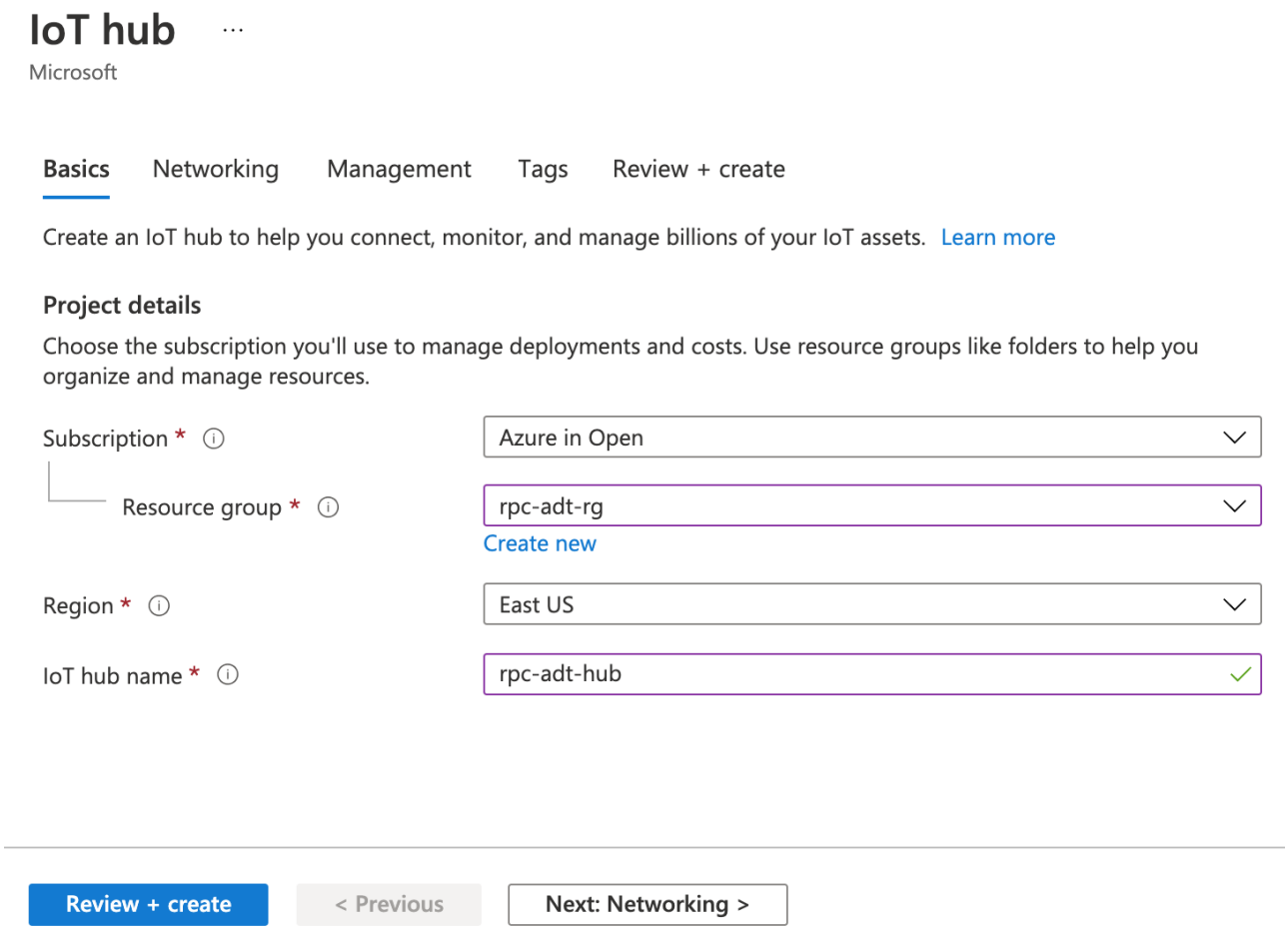


- 8.4 Input data

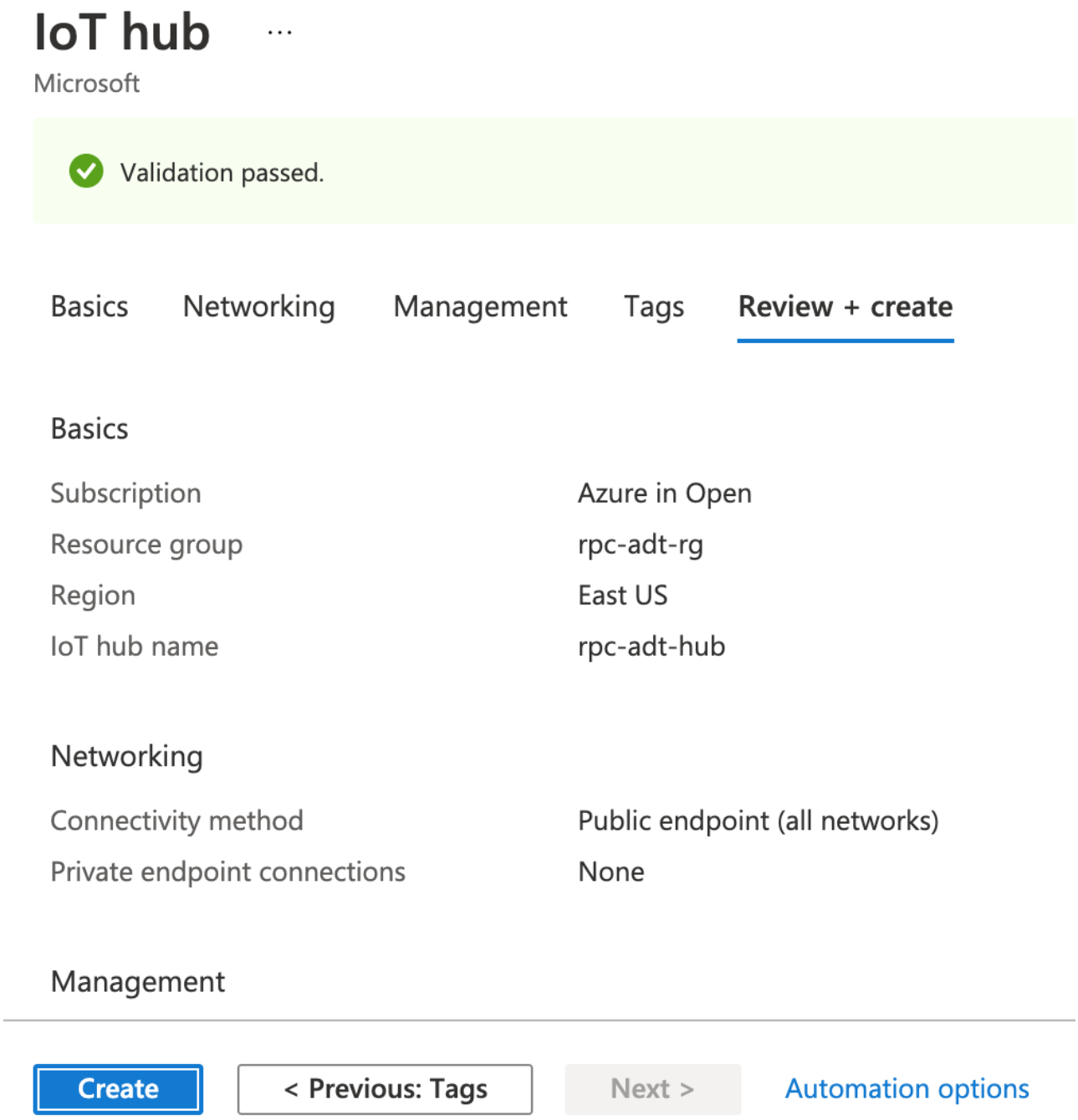


  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name, this example uses `rpc-adt-rg`
  - `Region` field
    - This example uses `East US`
  - `IoT hub name` field
    - Please use a recognizable name, this example uses `rpc-adt-hub`

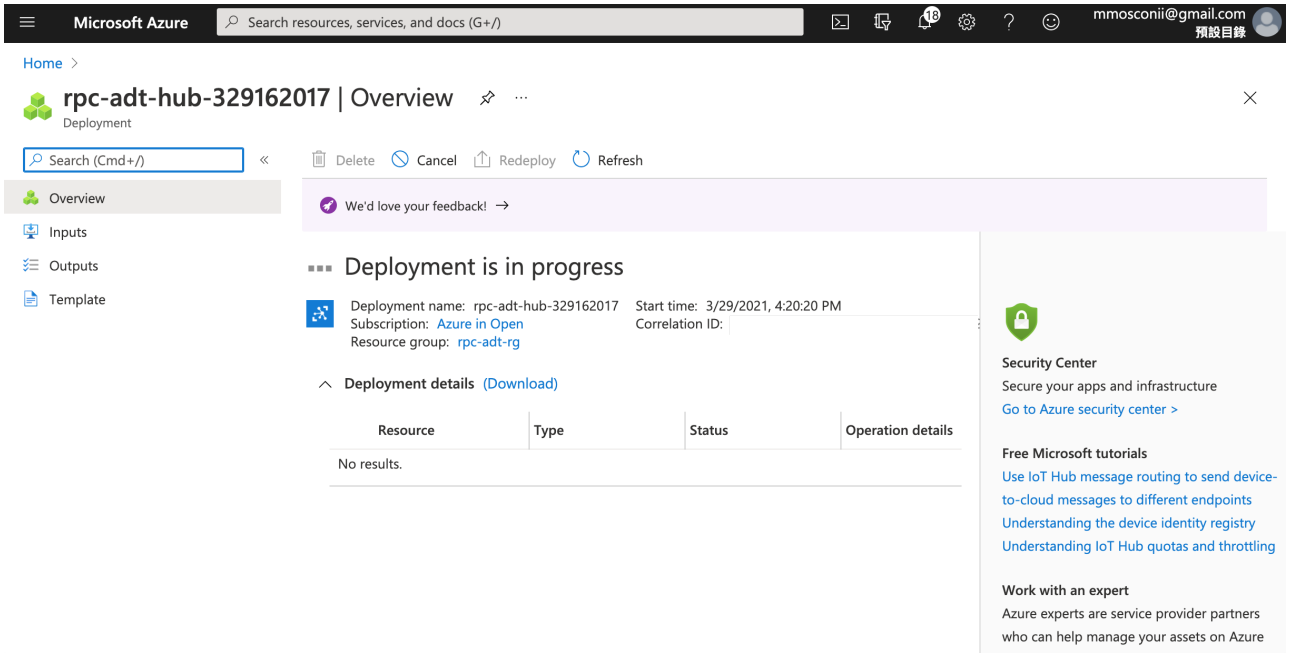
- 8.5 When the input is complete, please click the `Review + create` button



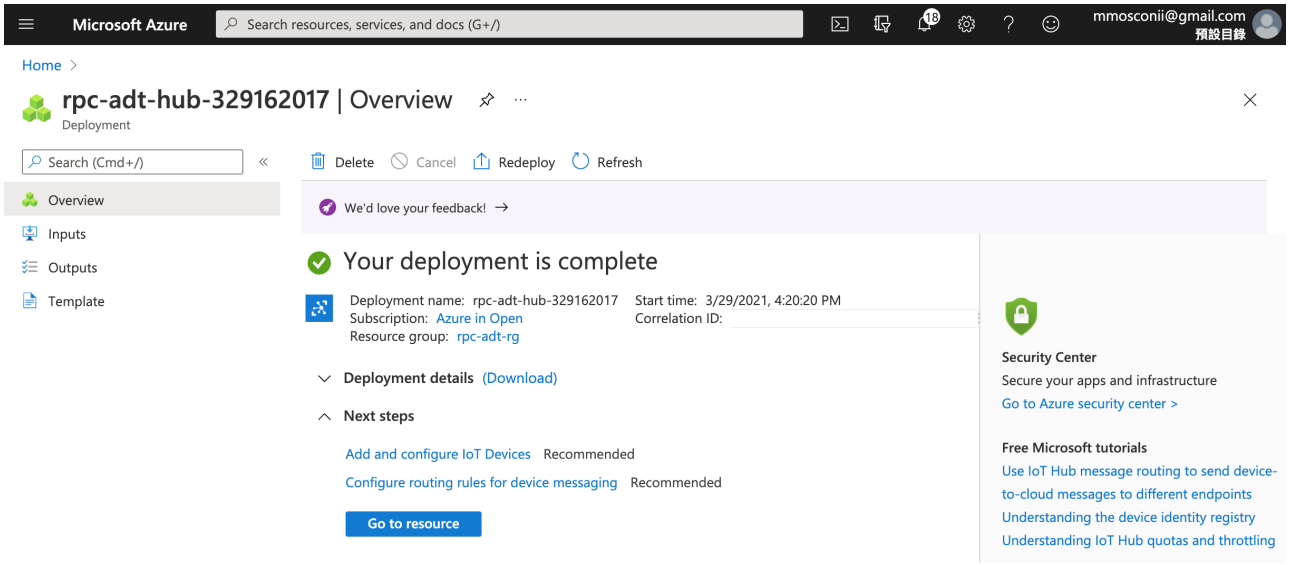
- 8.6 Review your settings and select `create`



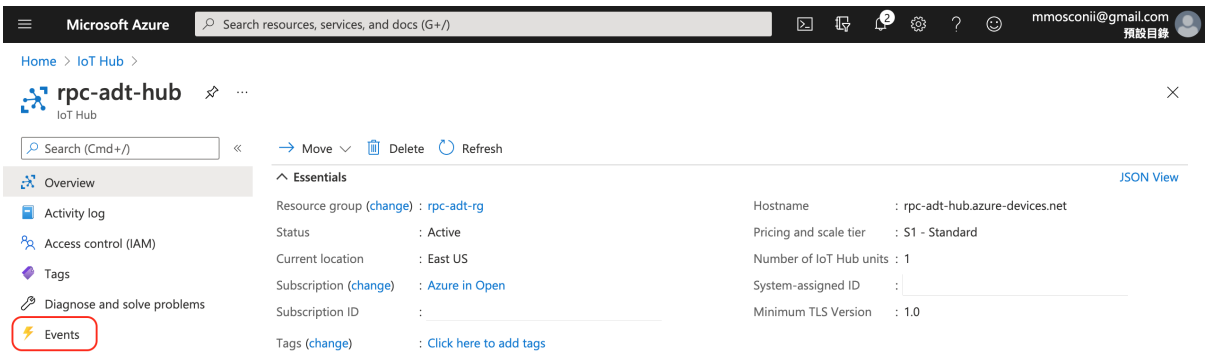
- 8.7 Wait for the request to process



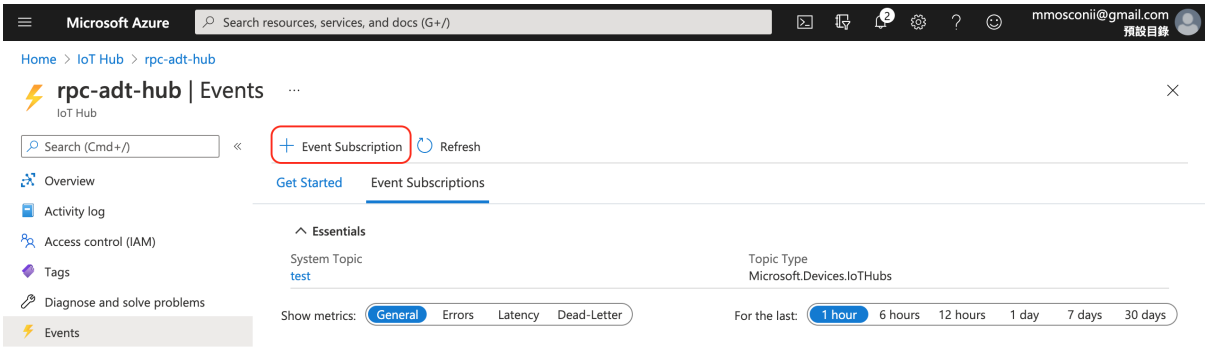
- 8.8 Operation completes



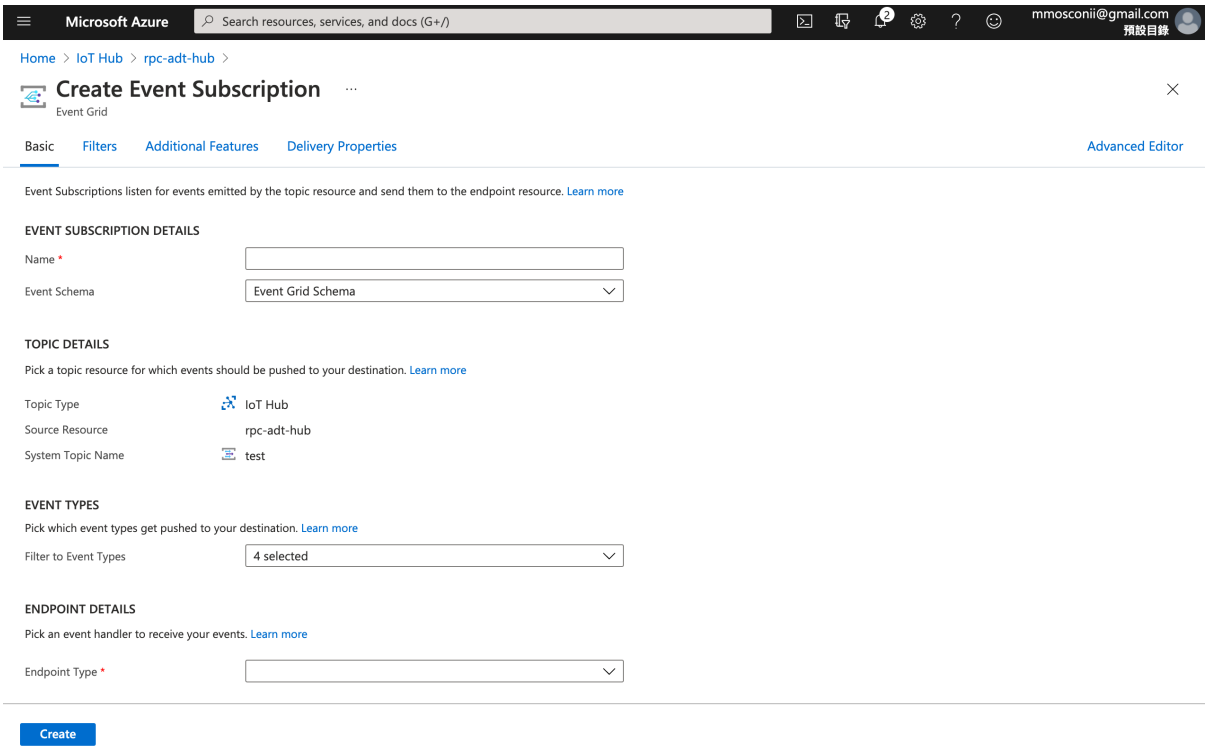
- 8.9 Set Event to `IngestADTFunctions` Azure Function
  - Click `Events`



- Click `+ Event Subscription`

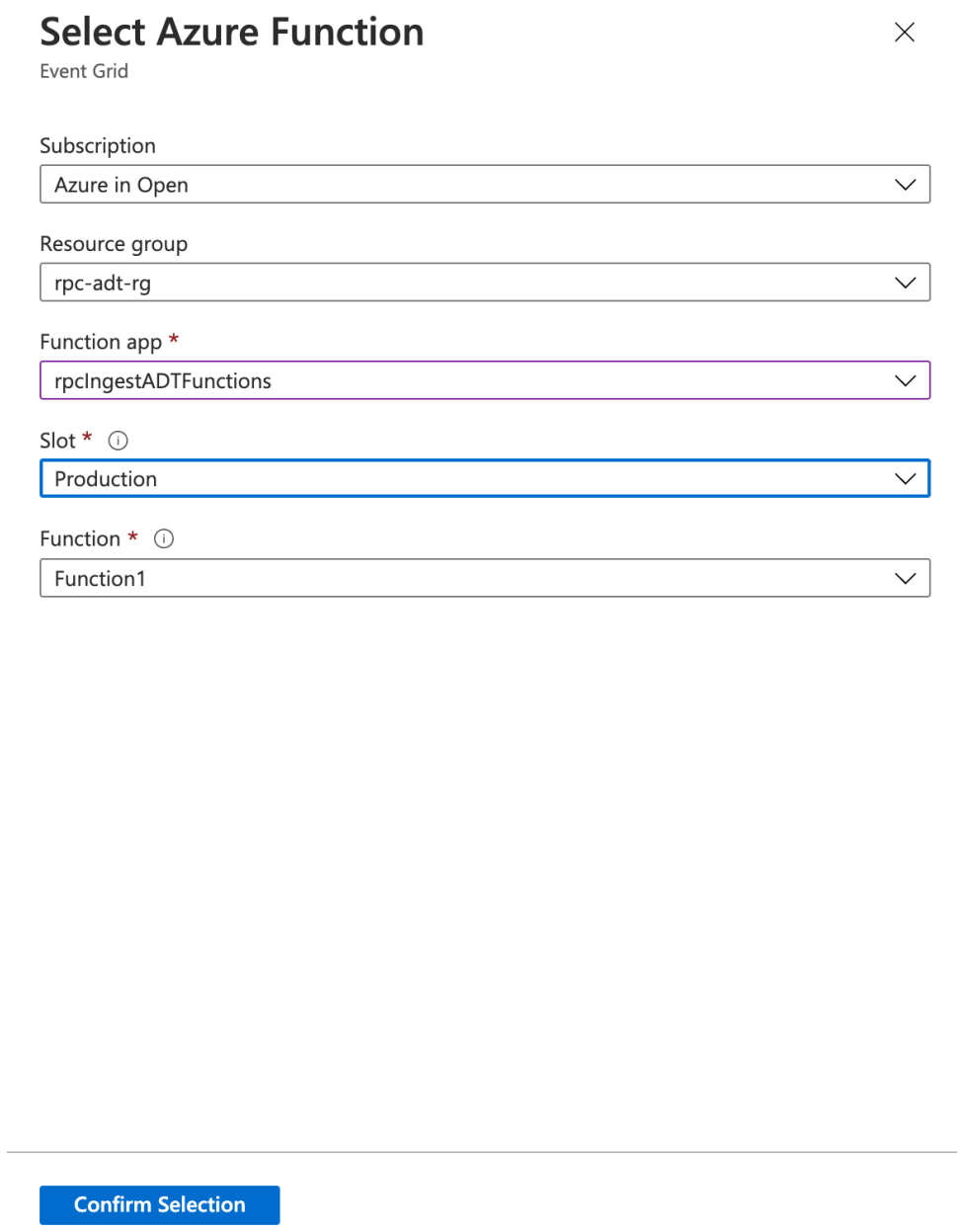


- Input data

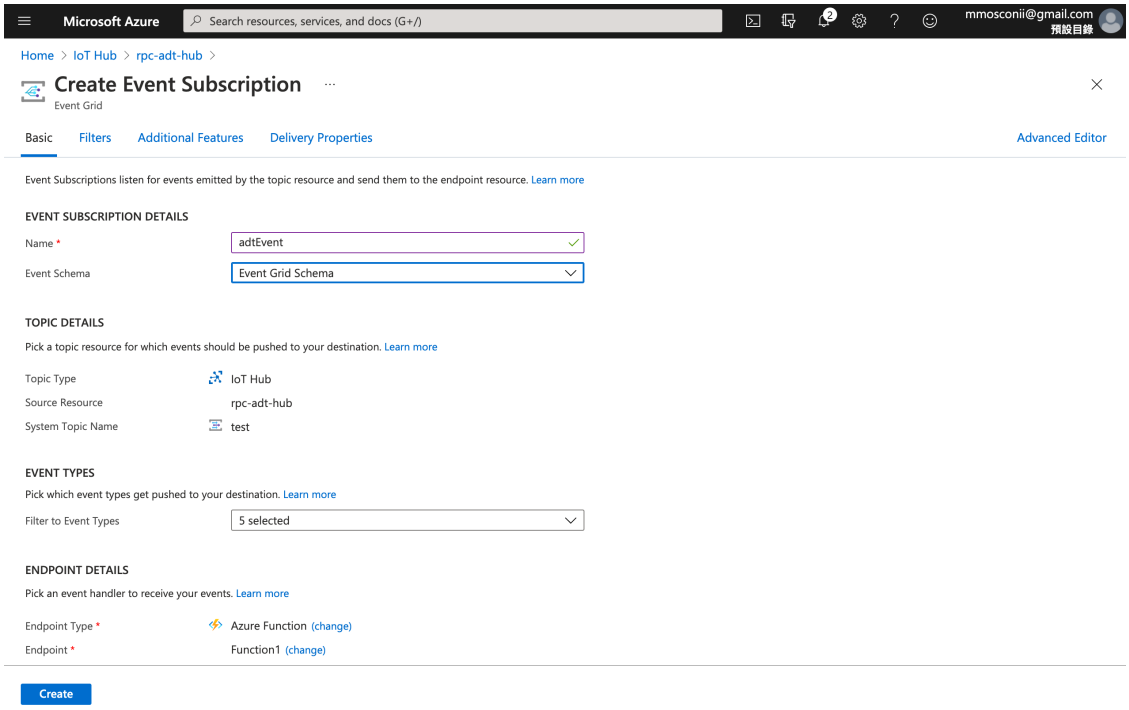


  - `Name` field
    - Please use a recognizable name, this example uses `adtEvent`
  - `System Topic Name` field
    - Please use a recognizable name, this example uses `test`
  - `Filter to Event Types` field
    - Please select `Device Created`、`Device Deleted`、`Device Connected`、`Device Disconnected`、`Device Telemetry`
  - `Endpoint Type` field
    - Please select `Azure Function`
  - `Endpoint` field
    - Please select `Select an endpoint`
    - `Select Azure Function` View
      - `Subscription` field
        - Select the subscription you want to use
      - `Resource group` field
        - Select your resource group from the dropdown, this example uses `rpc-adt-rg`
      - `Function app` field

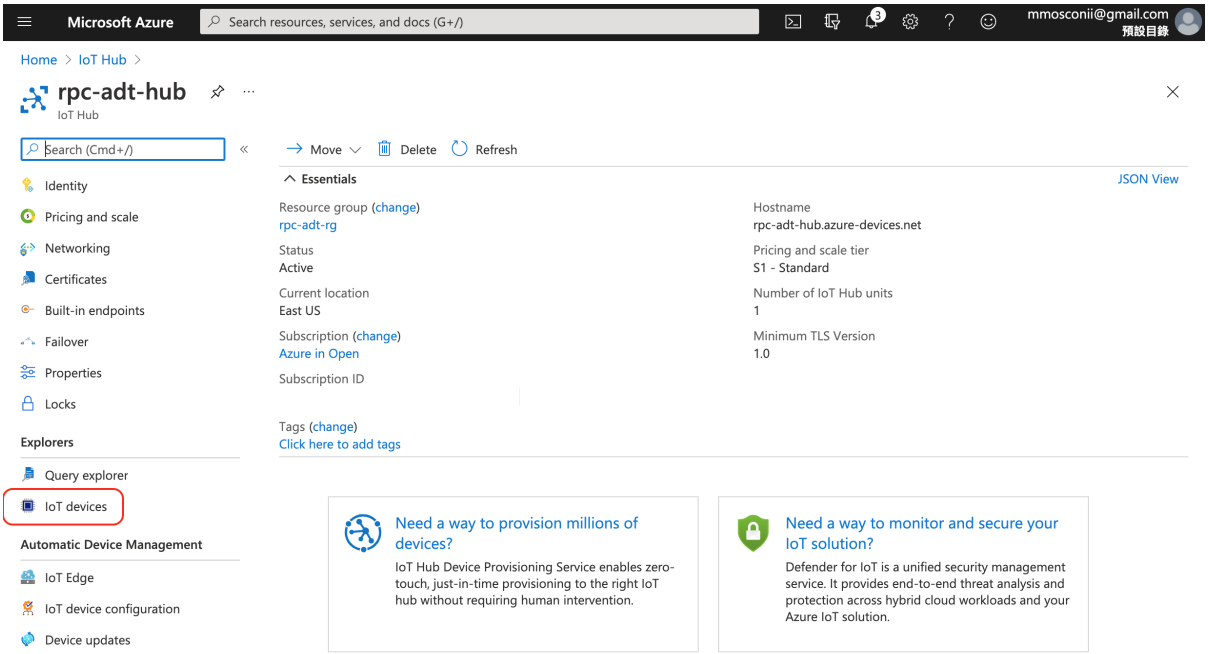
    - Select your azure function from the dropdown, this example uses `IngestADTFunctions`
- `Slot` field
    - Please select `Production`
- `Function` field
    - Please select `Function1`
- When the input is complete, please click the `Confirm Selection` button



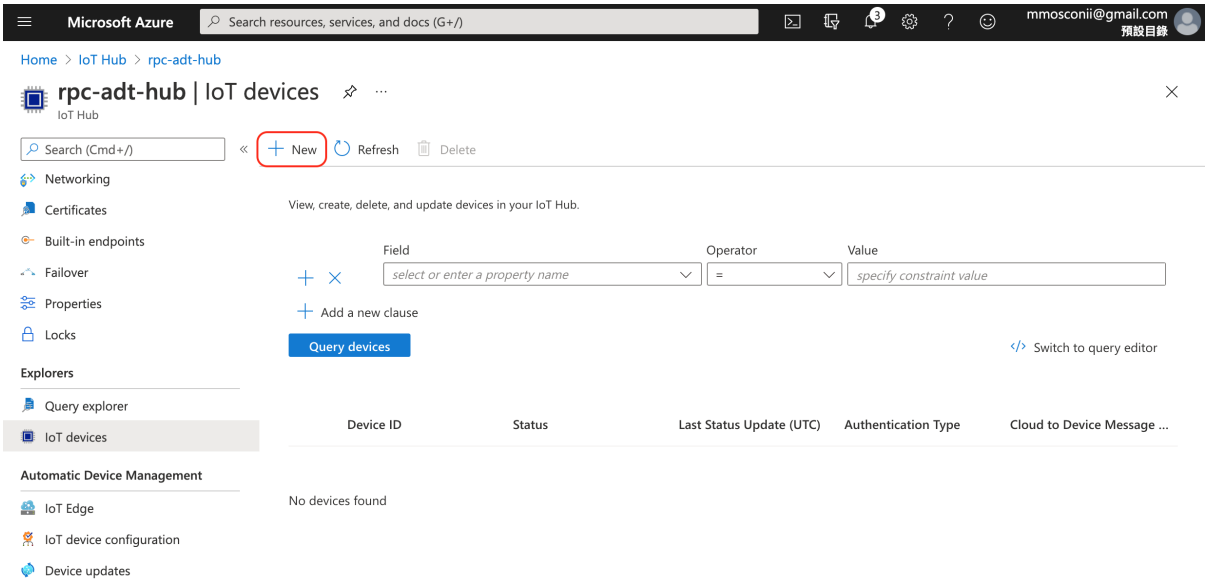
- When the input is complete, please click the `Create` button



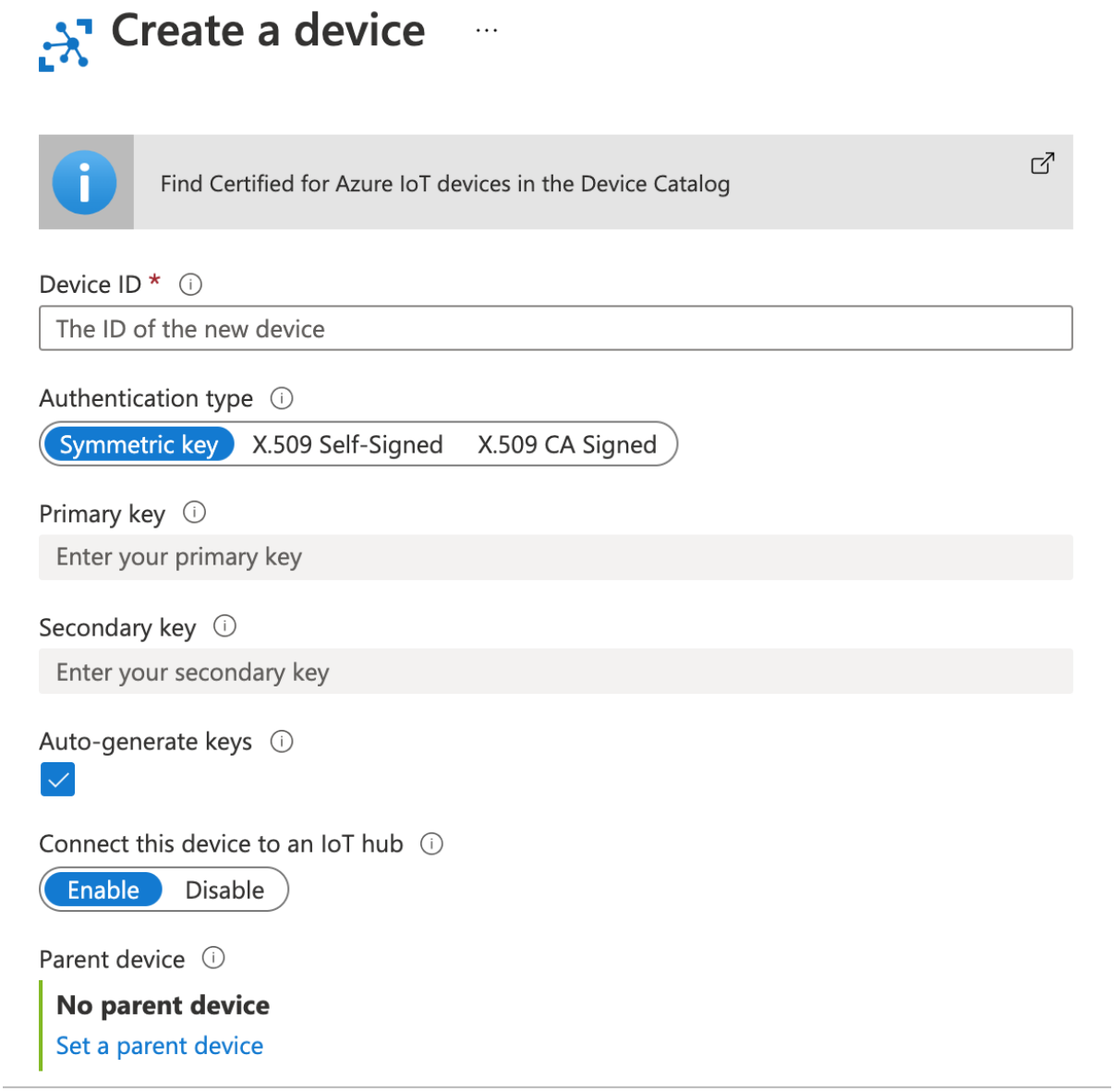
- 8.10 Create IoT Devices
  - Please create `rpc-adt-001`、`rpc-adt-002`、`rpc-adt-003`、`rpc-adt-004`
  - Click `IoT Devices`



- Click + New

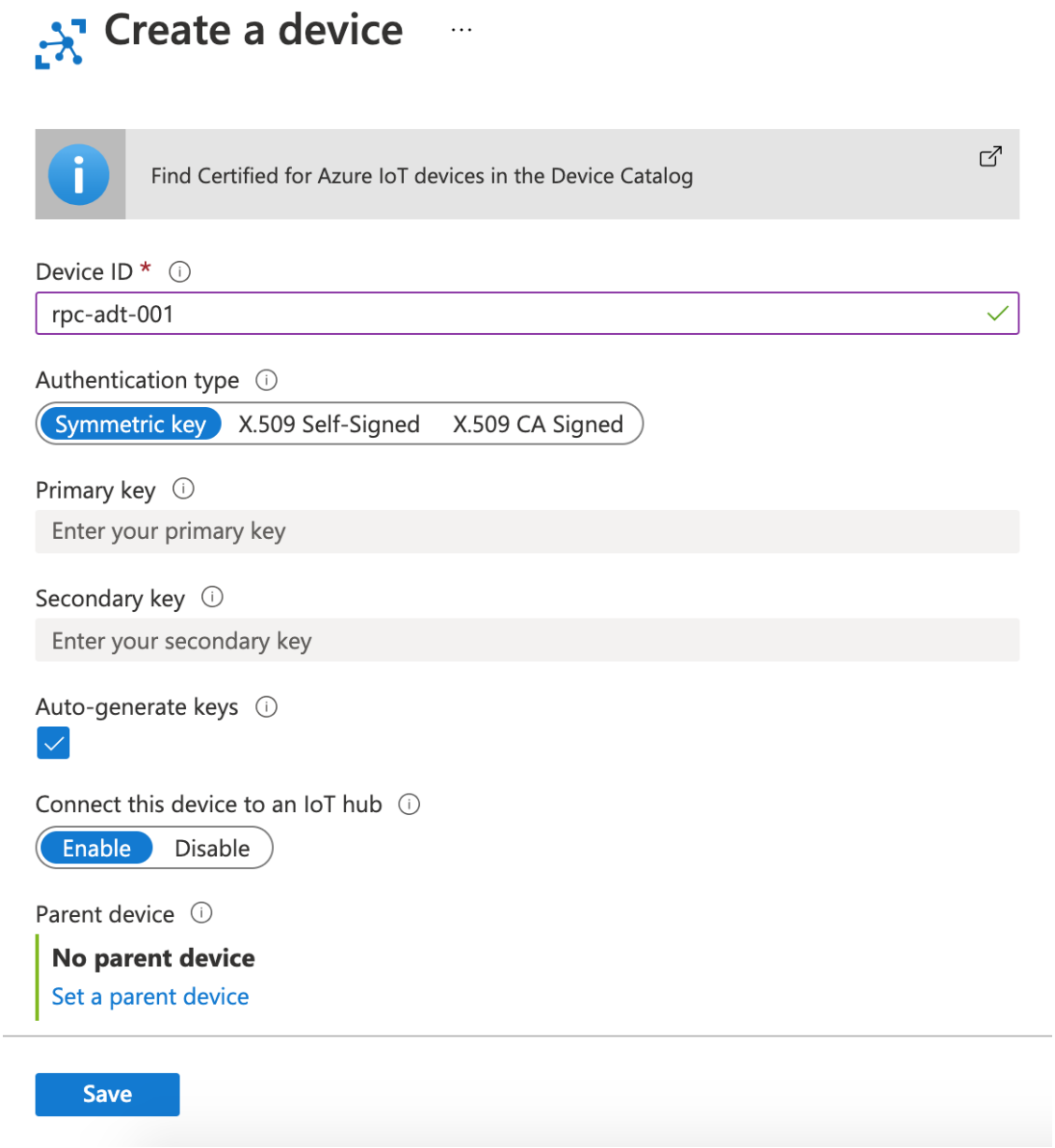


- Input data

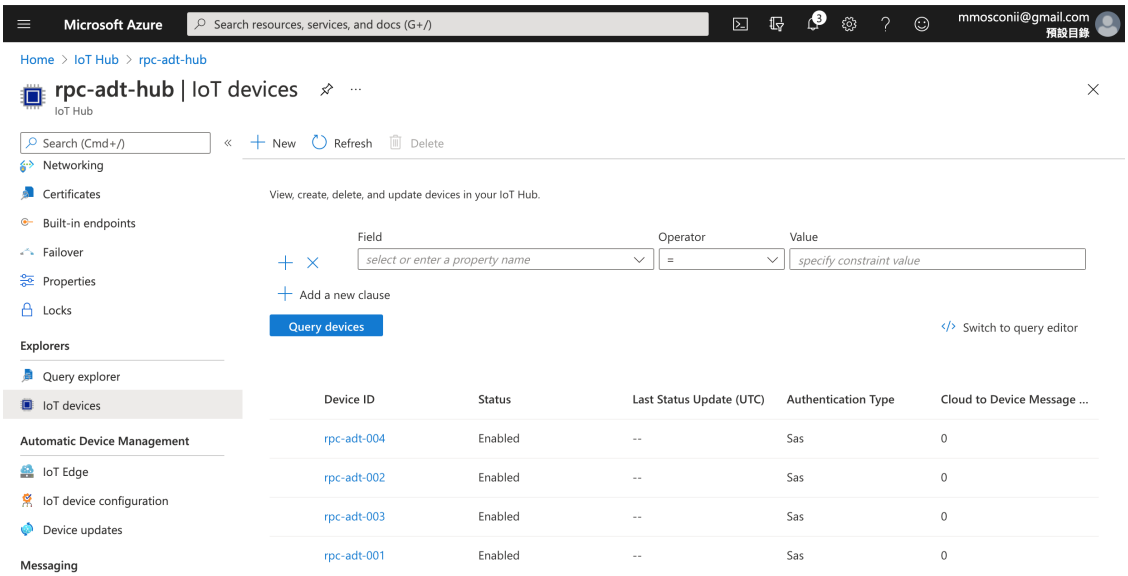


  - `Device ID` field

    - Input `rpc-adt-001`
- When the input is complete, please click the `Save` button



- Create `rpc-adt-002`、`rpc-adt-003`、`rpc-adt-004`



9. Set Event & toTsi Function

- 9.1 Create an event hub namespace that will receive events from your Azure Digital Twins instance
    - Azure CLI

```
az eventhubs namespace create --name <name for your Event Hubs
namespace> --resource-group <resource group name> -l <region>
```

    - e.g.

```
az eventhubs namespace create --name rpchubspace --resource-group
rpc-adt-rg -l eastus
```

- 9.2 Create an event hub within the namespace to receive twin change events. Specify a name for the event hub.
    - Azure CLI

```
az eventhubs eventhub create --name <name for your Twins event
hub> --resource-group <resource group name> --namespace-name
<Event Hubs namespace from above>
```

    - e.g.

```
az eventhubs eventhub create --name rpcadteventhub --resource-
group rpc-adt-rg --namespace-name rpchubspace
```

- 9.3 Create an authorization rule with send and receive permissions. Specify a name for the rule.
    - Azure CLI

```
az eventhubs eventhub authorization-rule create --rights Listen
Send --resource-group <resource group name> --namespace-name
<Event Hubs namespace from above> --eventhub-name <Twins event
hub name from above> --name <name for your Twins auth rule>
```

    - e.g.

```
az eventhubs eventhub authorization-rule create --rights Listen
Send --resource-group rpc-adt-rg --namespace-name rpchubspace --
eventhub-name rpcadteventhub --name rpcadteventrules
```

- 9.4 Create an Azure Digital Twins endpoint that links your event hub to your Azure Digital Twins instance.
    - Azure CLI

```
az dt endpoint create eventhub -n <your Azure Digital Twins
instance name> --endpoint-name <name for your Event Hubs
endpoint> --eventhub-resource-group <resource group name> --
eventhub-namespace <Event Hubs namespace from above> --eventhub
<Twins event hub name from above> --eventhub-policy <Twins auth
rule from above>
```

    - e.g.

```
az dt endpoint create eventhub --endpoint-name rpchubadtendpoint
--eventhub-resource-group rpc-adt-rg --eventhub-namespace
rpchubspace --eventhub rpcadteventhub --eventhub-policy
rpcadteventrules -n rpc-adt-example --resource-group rpc-adt-rg
```

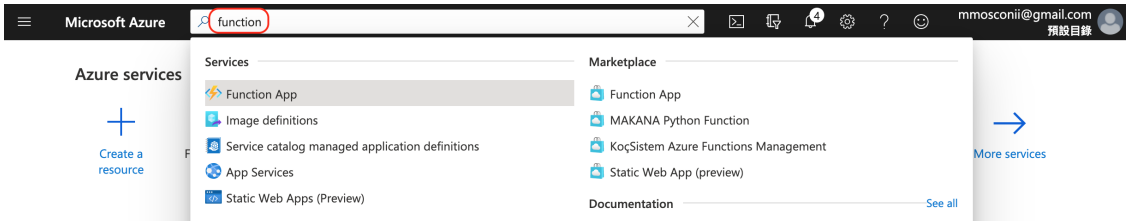
- 9.5 Create a route in Azure Digital Twins to send twin update events to your endpoint. The filter in this route will only allow twin update messages to be passed to your endpoint.
    - Azure CLI

```
az dt route create -n <your Azure Digital Twins instance name> --
endpoint-name <Event Hub endpoint from above> --route-name <name
for your route> --filter "type =
'Microsoft.DigitalTwins.Twin.Update'"
```

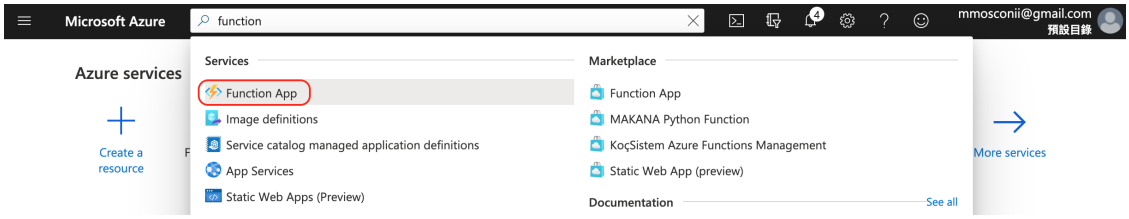
    - e.g.

```
az dt route create -n rpc-adt-example --endpoint-name
rpchubadtendpoint --route-name rpcroutename --filter "type =
'Microsoft.DigitalTwins.Twin.Update'" --resource-group rpc-adt-rg
```

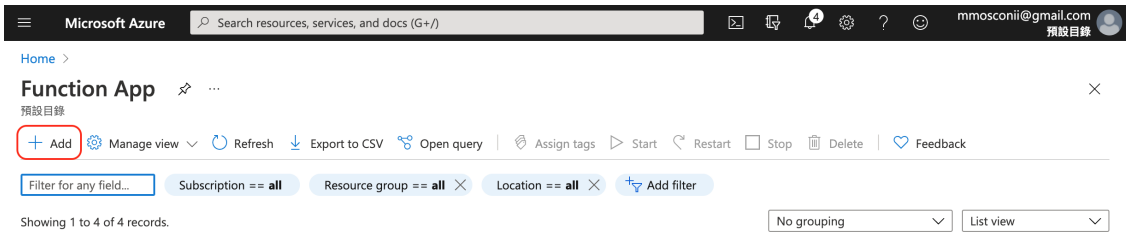
- 9.6 Create & Deploy a function in Azure
    - 8.6.1 Create Azure Functions
        - Search `function`



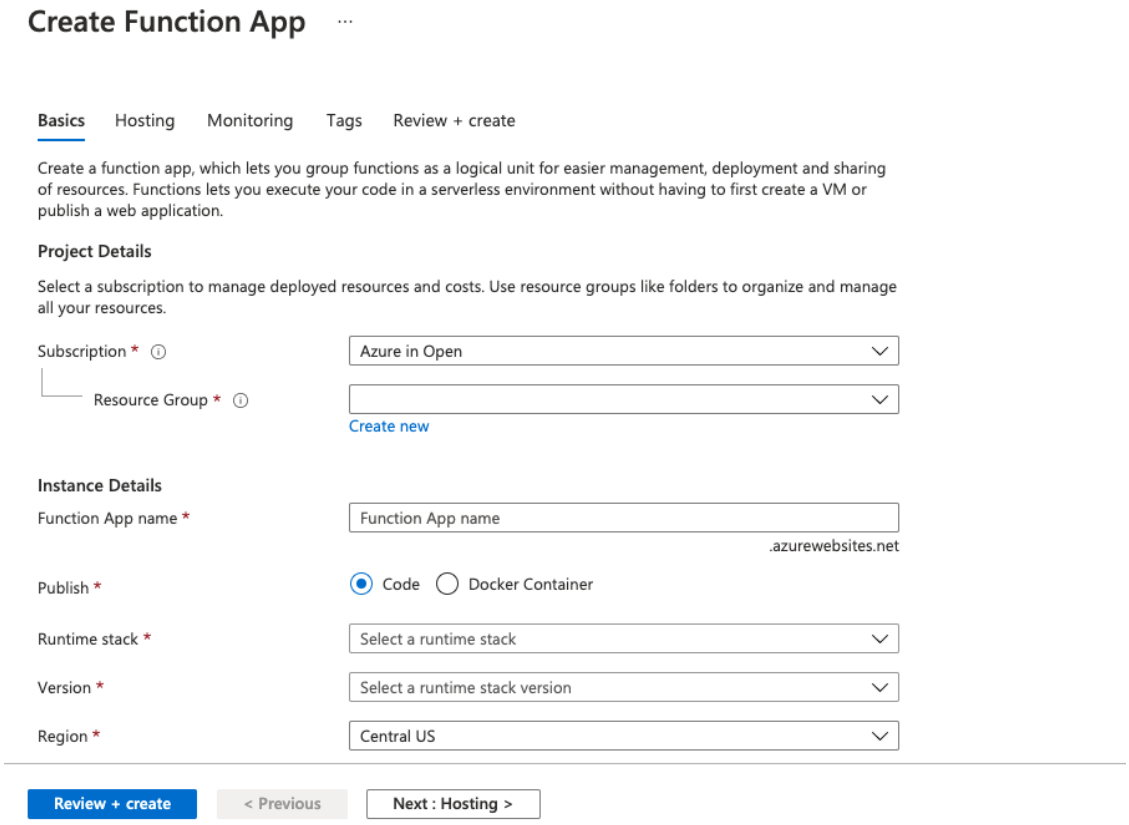
- Select `Function App`



- Click `Add`

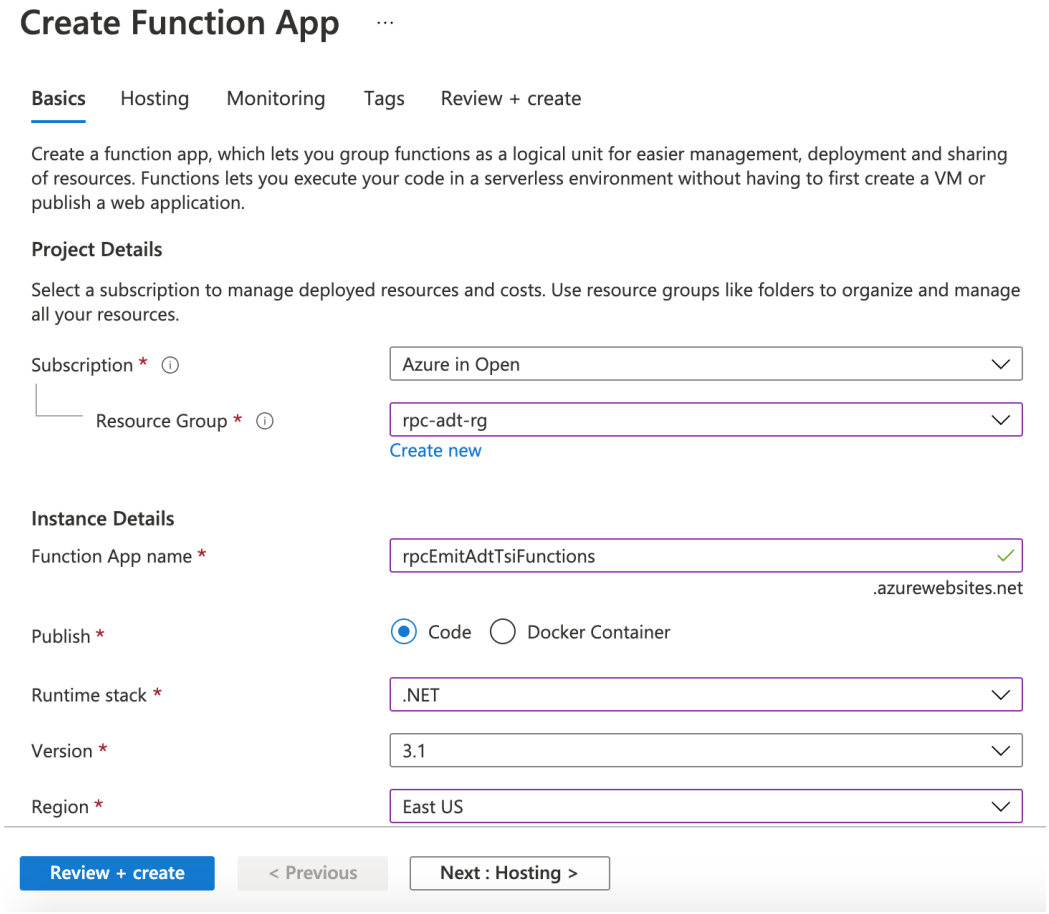


- Input data

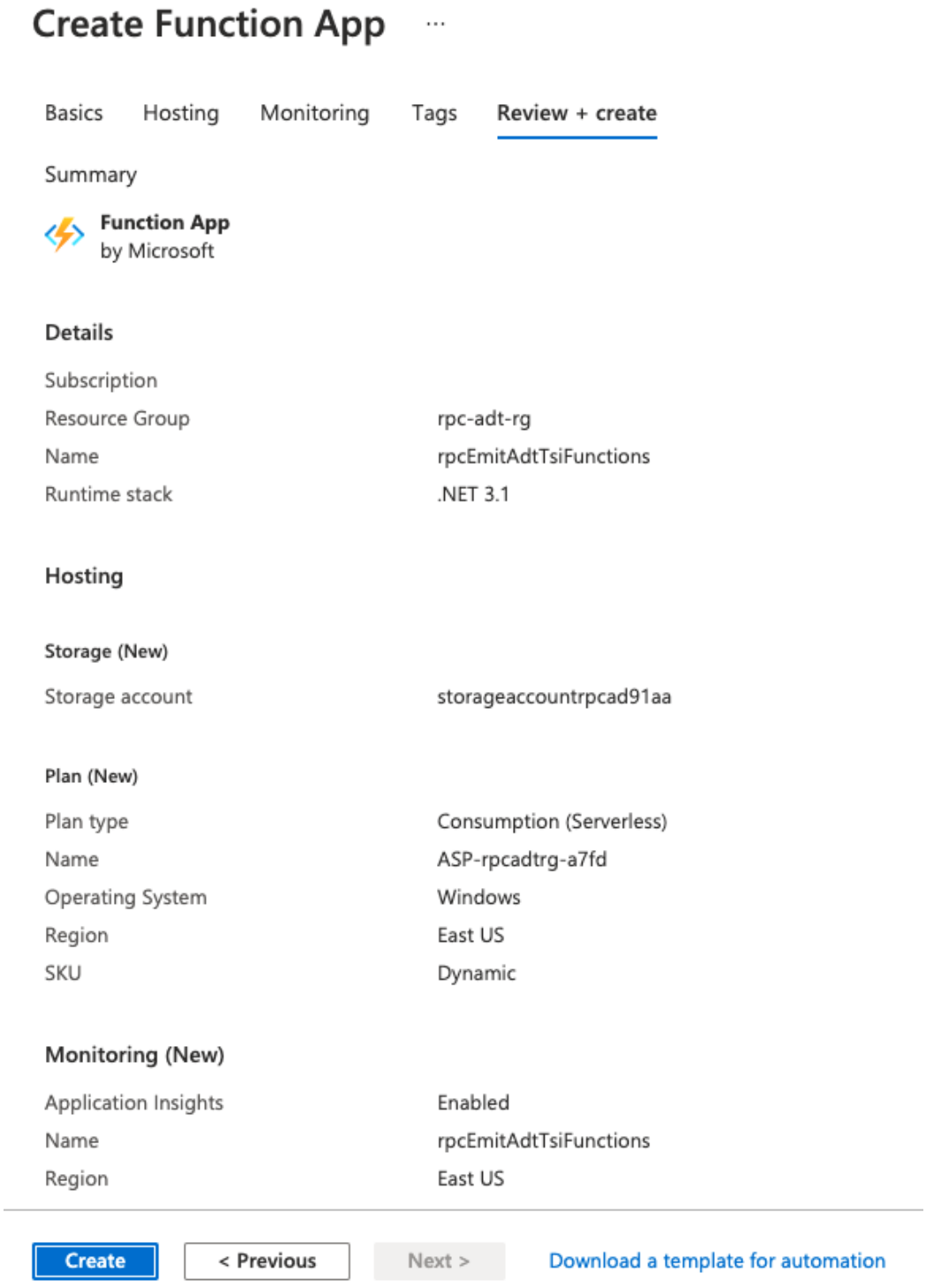


  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name, this example uses `rpc-adt-rg`
  - `Function App name` field
    - Please use a recognizable name, this example uses `rpcEmitAdtTsiFunctions`
  - `Publish` field
    - Select `Code`
  - `Runtime stack` field
    - Select `.NET`
  - `Version` field
    - Select `3.1`

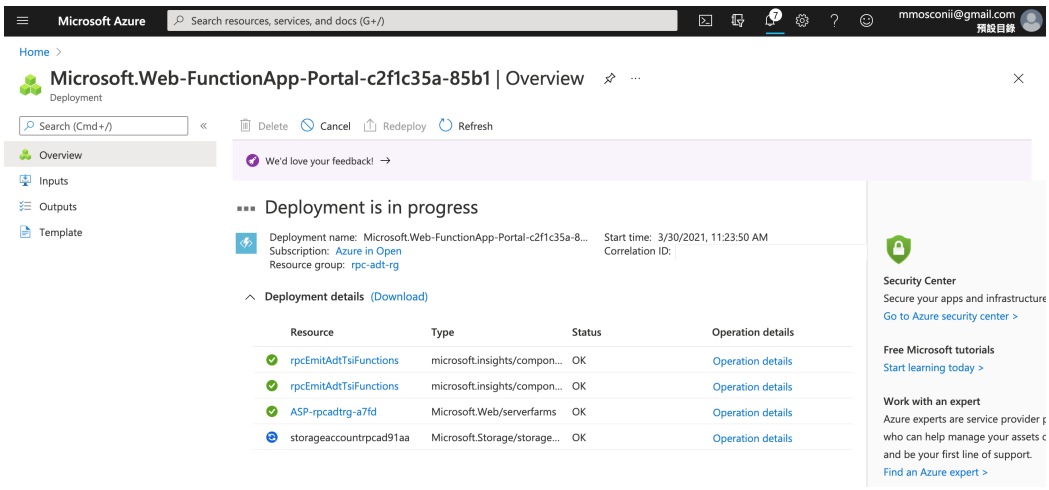
- `Region` field
  - This example uses `East US`
- When the input is complete, please click the `Review + create` button



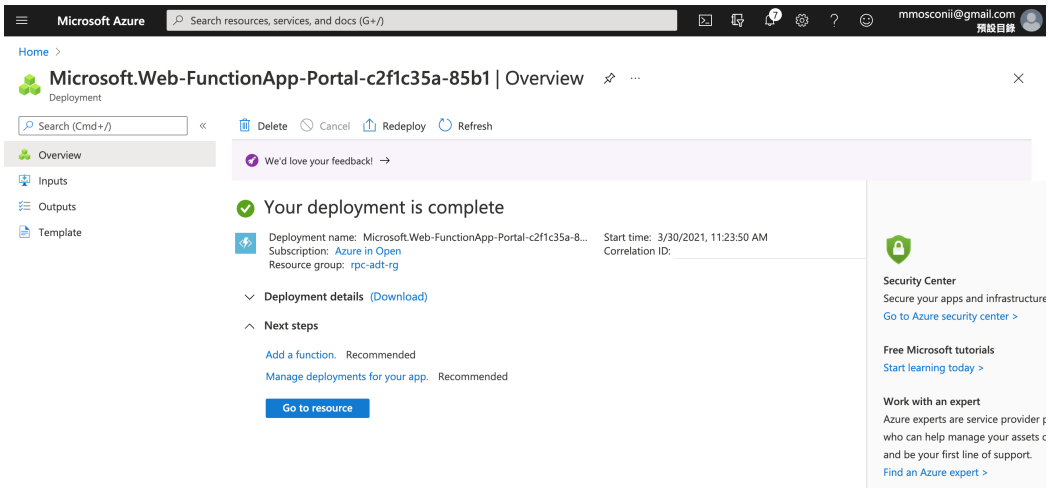
- Review your settings and select `Create`



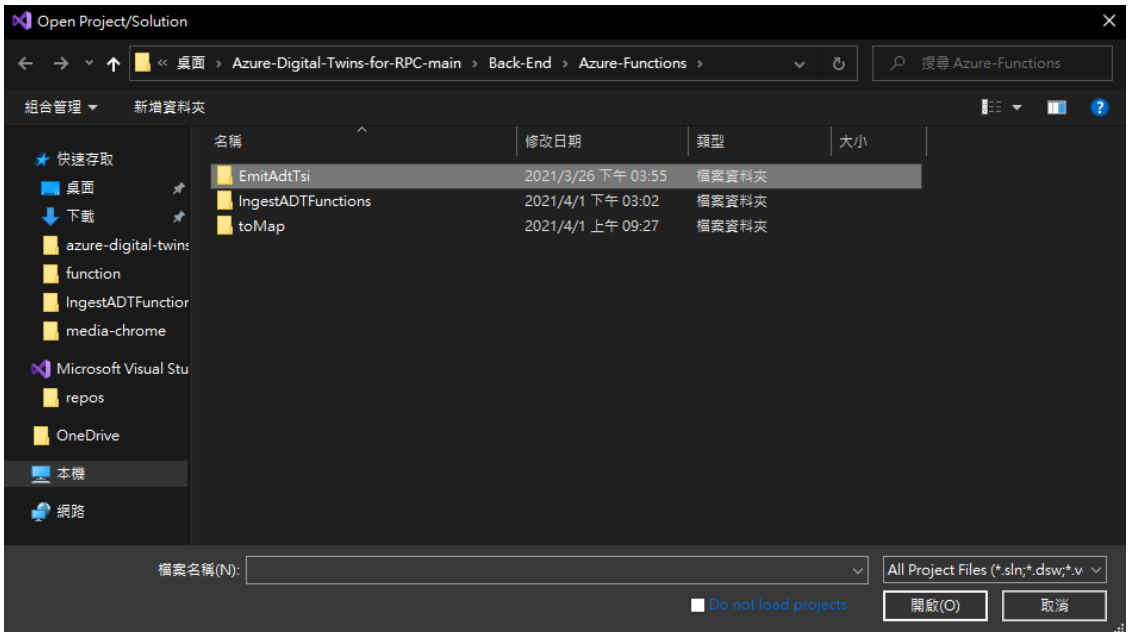
- wait for the request to process

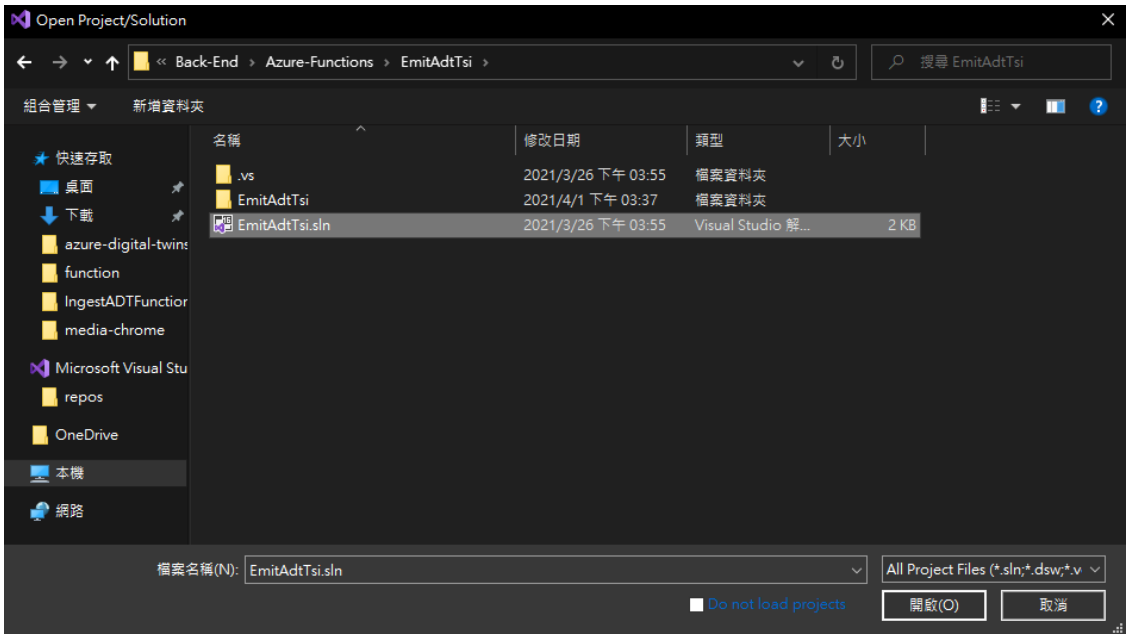


- Once deployment complete click on `Go to resource` button

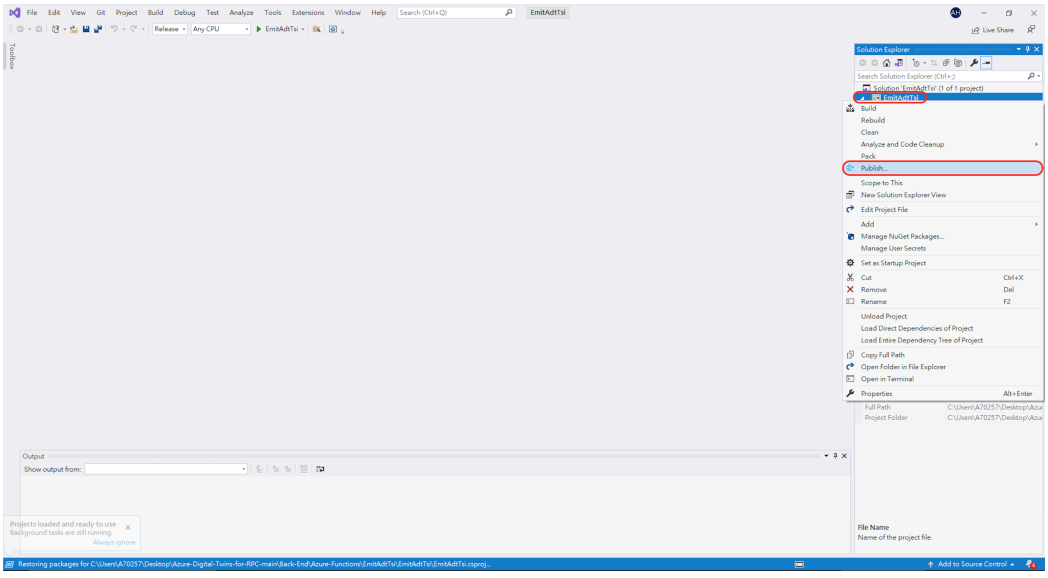


- 9.6.2 Deploy
  - Opening `EmitAdtTsi.sln` from `/Back-End/Azure-Functions/EmitAdtTsi` in `Visual Studio 2019`

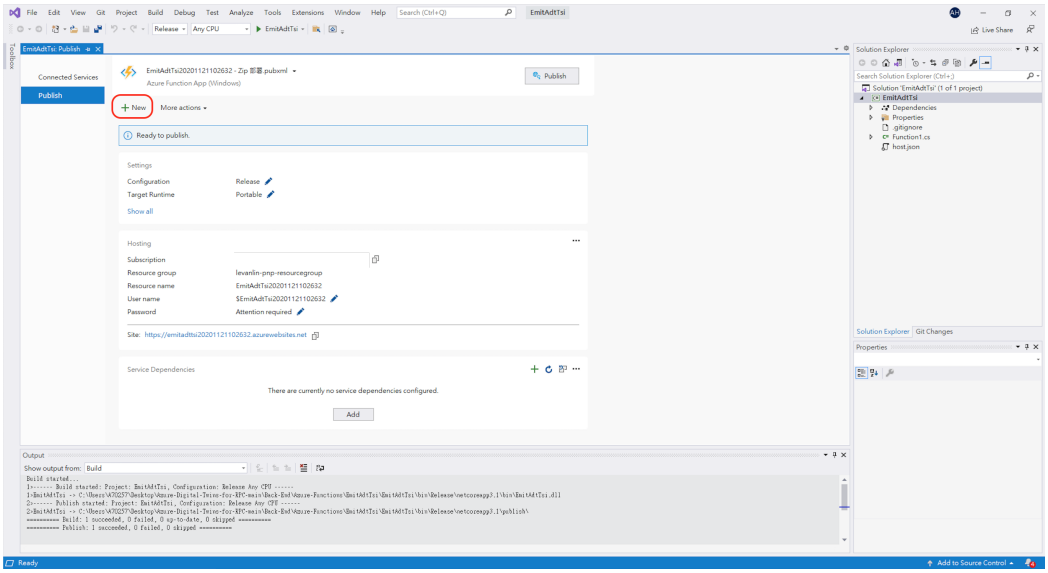




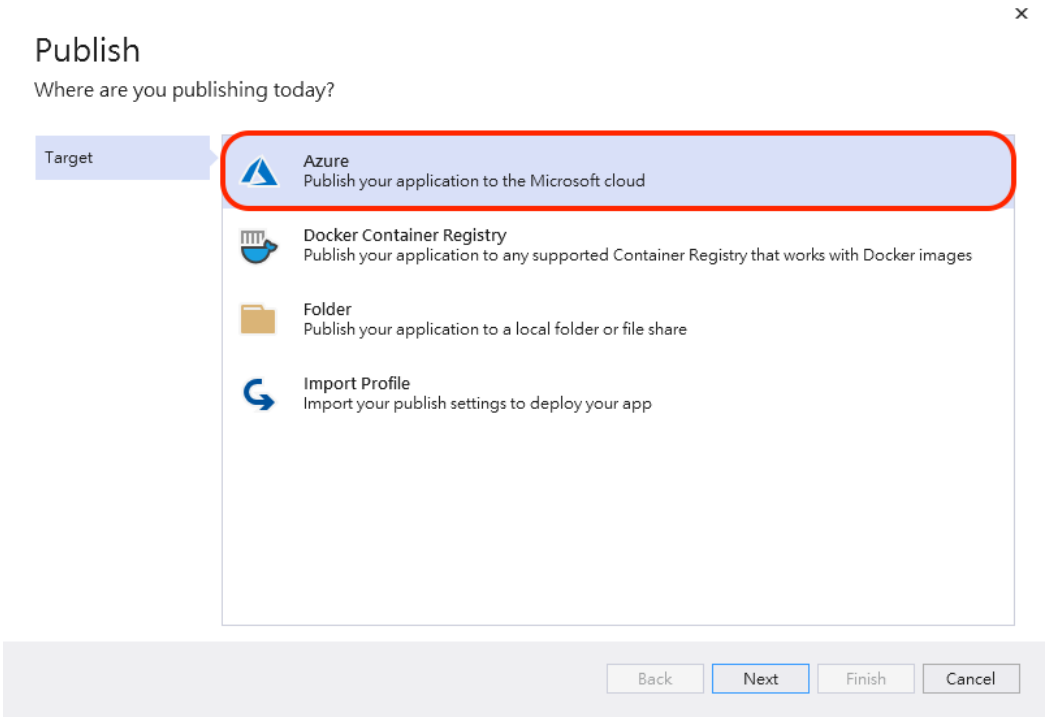
- Deploy `EmitAdtTsi` to `Azure Function`
  - Click `Publish...`



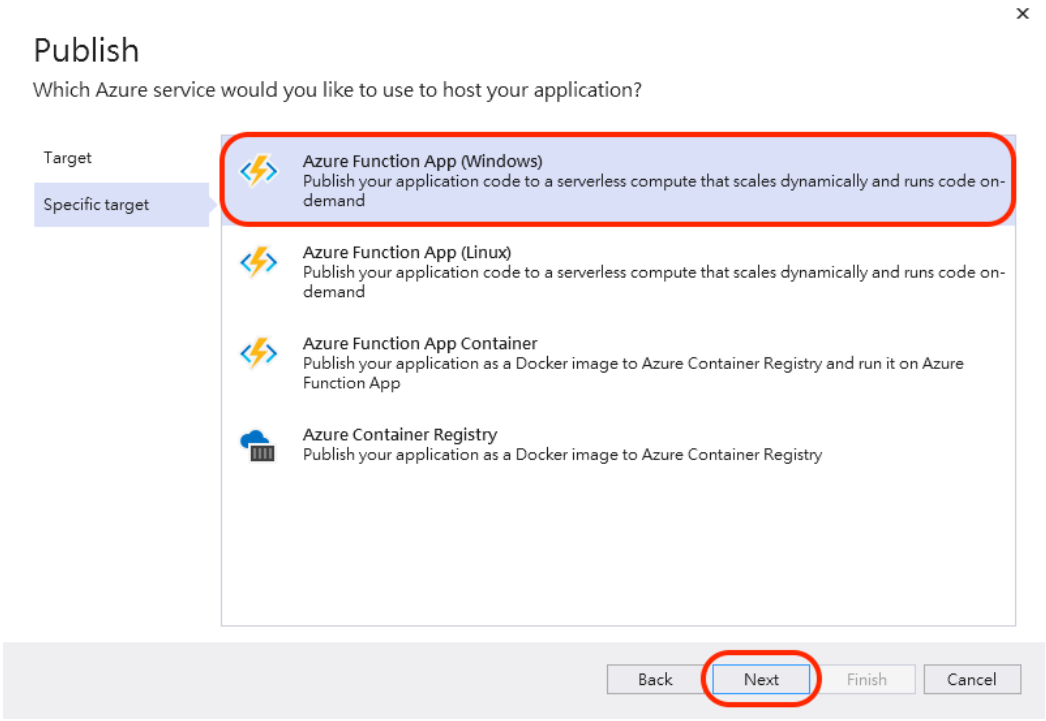
- Click + New



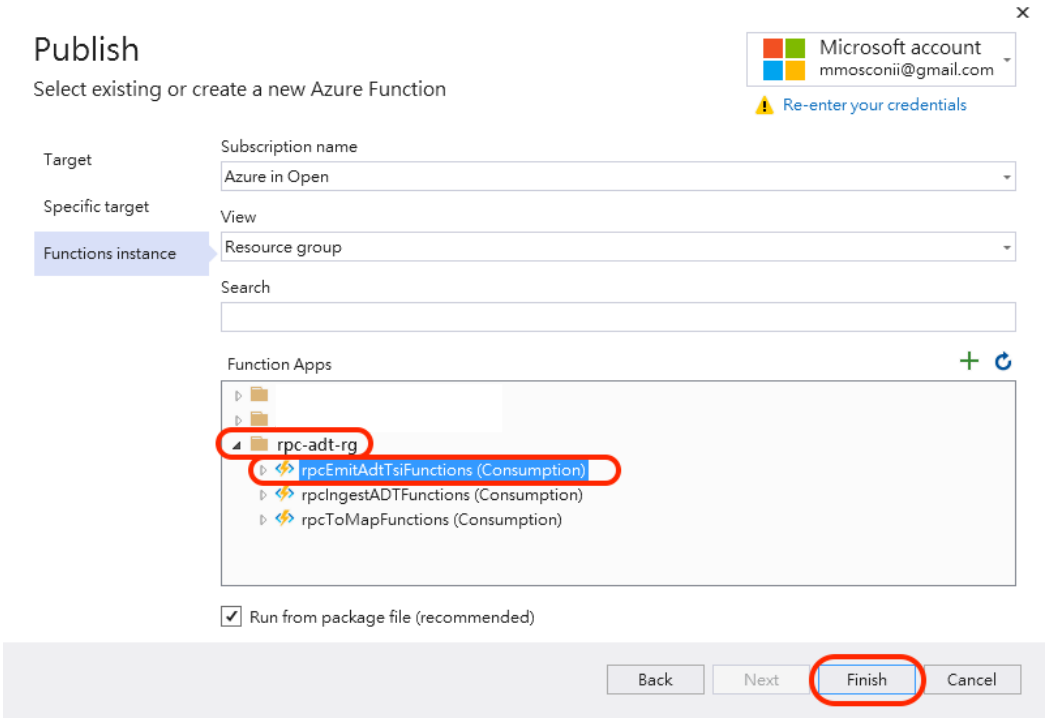
- Click Azure



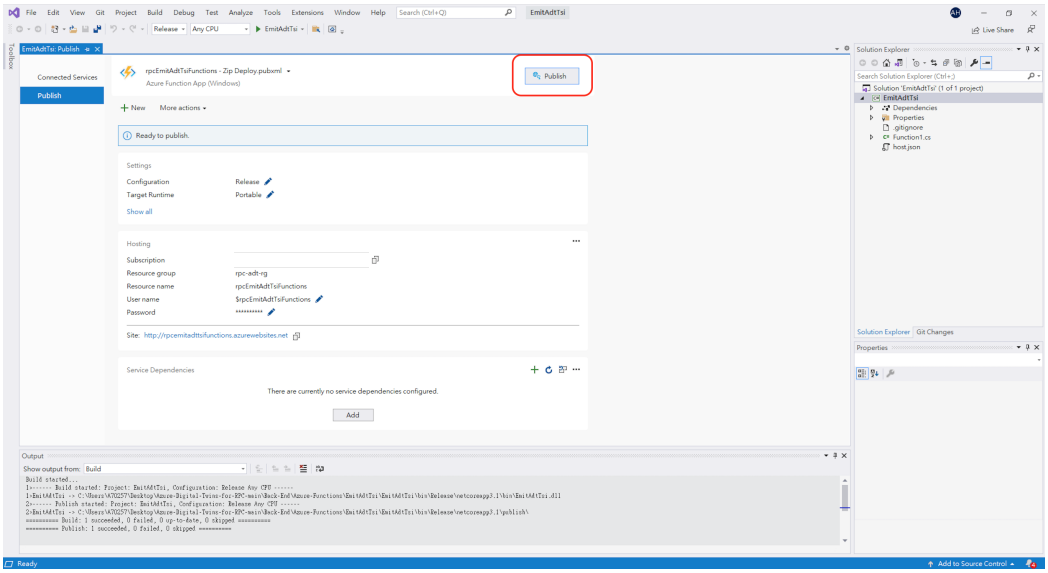
- Click `Azure Function App (Windows)` > `Next`



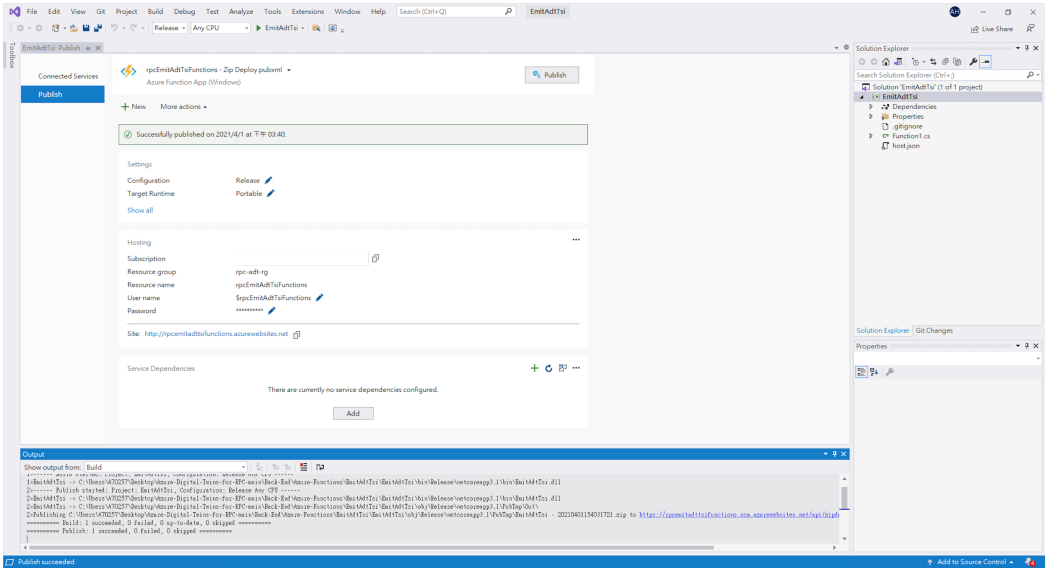
- Please select the function created in `Step 9.6.1`



- Click Publish



- Operation completes



- 9.7 Create a new event hub. Specify a name for the event hub.
  - Azure CLI

```
az eventhubs eventhub create --name <name for your TSI event hub>
--resource-group <resource group name from earlier> --namespace-
name <Event Hubs namespace from earlier>
```

  - e.g.

```
az eventhubs eventhub create --name rptsieventhub --resource-
group rpc-adt-rg --namespace-name rpchubspace
```

- 9.8 Create an authorization rule with send and receive permissions. Specify a name for the rule.

- Azure CLI

```
az eventhubs eventhub authorization-rule create --rights Listen
Send --resource-group <resource group name> --namespace-name
<Event Hubs namespace from earlier> --eventhub-name <TSI event
hub name from above> --name <name for your TSI auth rule>
```

- e.g.

```
az eventhubs eventhub authorization-rule create --rights Listen
Send --resource-group rpc-adt-rg --namespace-name rpchubspace --
eventhub-name rptsieventhub --name rpctsieventrules
```

- 9.9 Get the Twins event hub connection string, using the authorization rules you created above for the Twins hub.
  - Azure CLI

```
az eventhubs eventhub authorization-rule keys list --resource-
group <resource group name> --namespace-name <Event Hubs
namespace> --eventhub-name <Twins event hub name from earlier> --
name <Twins auth rule from earlier>
```

  - e.g.

```
az eventhubs eventhub authorization-rule keys list --resource-
group rpc-adt-rg --namespace-name rpchubspace --eventhub-name
rpcadteventhub --name rpcadteventrules
```

- 9.10 Get `Twins event hub connection string` (primaryConnectionString)
  - Azure CLI

```
az eventhubs eventhub authorization-rule keys list --resource-
group <resource group name> --namespace-name <Event Hubs
namespace> --eventhub-name <Twins event hub name> --name <Twins
auth rule>
```

  - e.g.

```
az eventhubs eventhub authorization-rule keys list --resource-
group rpc-adt-rg --namespace-name rpchubspace --eventhub-name
```

```
rpcadteventhub --name rpcadteventrules
```

- Result (To get `primaryConnectionString`)

```
{
  "aliasPrimaryConnectionString": null,
  "aliasSecondaryConnectionString": null,
  "keyName": "rpcadteventrules",
  "primaryConnectionString":
"Endpoint=sb://rpchubspace.servicebus.windows.net/;SharedAccessKe
yName=rpcadteventrules;SharedAccessKey=LnJ4NE3V05pq6RYsY2Fpm5sCTr
9WGyuWd46yZ70fJA8=;EntityPath=rpcadteventhub",
  "primaryKey": "LnJ4NE3V05pq6RYsY2Fpm5sCTr9WGyuWd46yZ70fJA8=",
  "secondaryConnectionString":
"Endpoint=sb://rpchubspace.servicebus.windows.net/;SharedAccessKe
yName=rpcadteventrules;SharedAccessKey=FTsyYp7TEWOJcHCHNBUC55+Nyv
Fm7cdGpH+eLhMhqG4=;EntityPath=rpcadteventhub",
  "secondaryKey": "FTsyYp7TEWOJcHCHNBUC55+NyvFm7cdGpH+eLhMhqG4="
}
```

- 9.11 Use the `primaryConnectionString` value from the result to create an app setting in your function app that contains your connection string:
  - Azure CLI

```
az functionapp config appsettings set --settings
"EventHubAppSetting-Twins=<Twins event hub connection string>" -g
<resource group> -n <your App Service (function app) name>
```

  - e.g.

```
az functionapp config appsettings set --settings
"EventHubAppSetting-
Twins=Endpoint=Endpoint=sb://rpchubspace.servicebus.windows.net/;
SharedAccessKeyName=rpcadteventrules;SharedAccessKey=LnJ4NE3V05pq
6RYsY2Fpm5sCTr9WGyuWd46yZ70fJA8=;EntityPath=rpcadteventhub" -g
rpc-adt-rg -n rpcEmitAdtTsiFunctions
```

- 9.12 Get the TSI event hub connection string, using the authorization rules you created above for the Time Series Insights hub
  - Azure CLI

```
az eventhubs eventhub authorization-rule keys list --resource-
group <resource group name> --namespace-name <Event Hubs
```

```
namespace> --eventhub-name <TSI event hub name> --name <TSI auth
rule>
```

- e.g.

```
az eventhubs eventhub authorization-rule keys list --resource-
group rpc-adt-rg --namespace-name rpchubspace --eventhub-name
rptsieventhub --name rpctsieventrules
```

- 9.13 Get `TSI event hub connection string` (primaryConnectionString)
  - Azure CLI

```
az eventhubs eventhub authorization-rule keys list --resource-
group <resource group name> --namespace-name <Event Hubs
namespace> --eventhub-name <TSI event hub name> --name <TSI auth
rule>
```

  - e.g.

```
az eventhubs eventhub authorization-rule keys list --resource-
group rpc-adt-rg --namespace-name rpchubspace --eventhub-name
rptsieventhub --name rpctsieventrules
```

  - Result (To get `primaryConnectionString`)

```
{
  "aliasPrimaryConnectionString": null,
  "aliasSecondaryConnectionString": null,
  "keyName": "rpctsieventrules",
  "primaryConnectionString":
"Endpoint=sb://rpchubspace.servicebus.windows.net/;SharedAccessKe
yName=rpctsieventrules;SharedAccessKey=xFE9i04ojQHDRQkZ4F8J6mIsWo
o9o4sAVyVbn/uv7tk=;EntityPath=rptsieventhub",
  "primaryKey": "xFE9i04ojQHDRQkZ4F8J6mIsWoo9o4sAVyVbn/uv7tk=",
  "secondaryConnectionString":
"Endpoint=sb://rpchubspace.servicebus.windows.net/;SharedAccessKe
yName=rpctsieventrules;SharedAccessKey=gS2UcdeTC+32FPBLc2O8dAGFWP
So/nA0F5WJXHRBkvA=;EntityPath=rptsieventhub",
  "secondaryKey": "gS2UcdeTC+32FPBLc2O8dAGFWPSo/nA0F5WJXHRBkvA="
}
```

- 9.14 Use the `primaryConnectionString` value from the result to create an app setting in your function app that contains your connection string

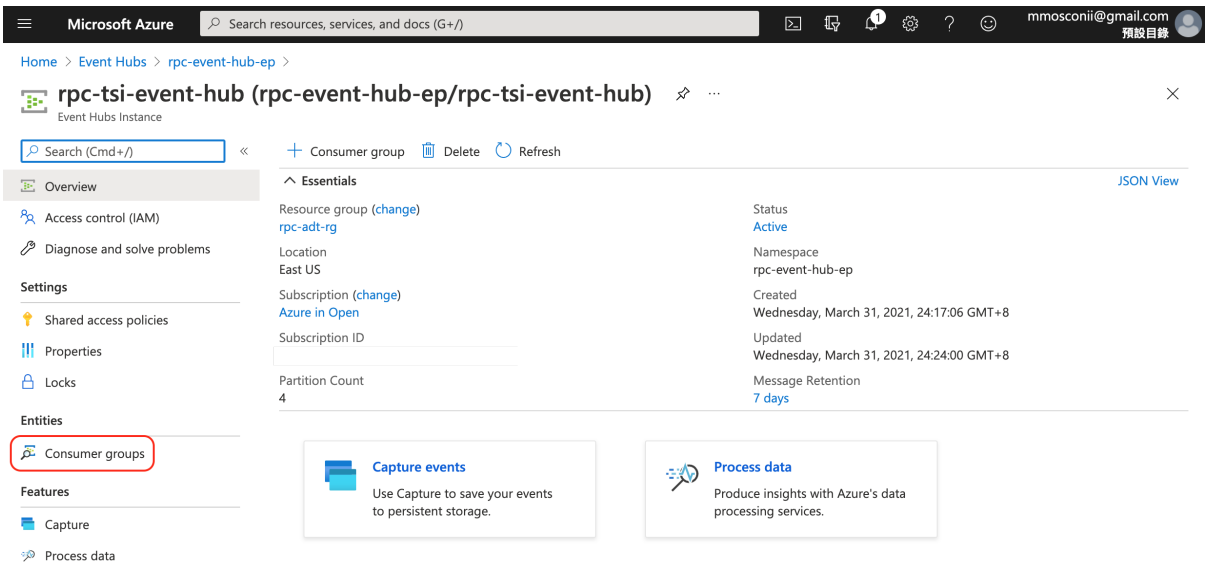
- Azure CLI

```
az functionapp config appsettings set --settings
"EventHubAppSetting-TSI=<TSI event hub connection string>" -g
<resource group> -n <your App Service (function app) name>
```

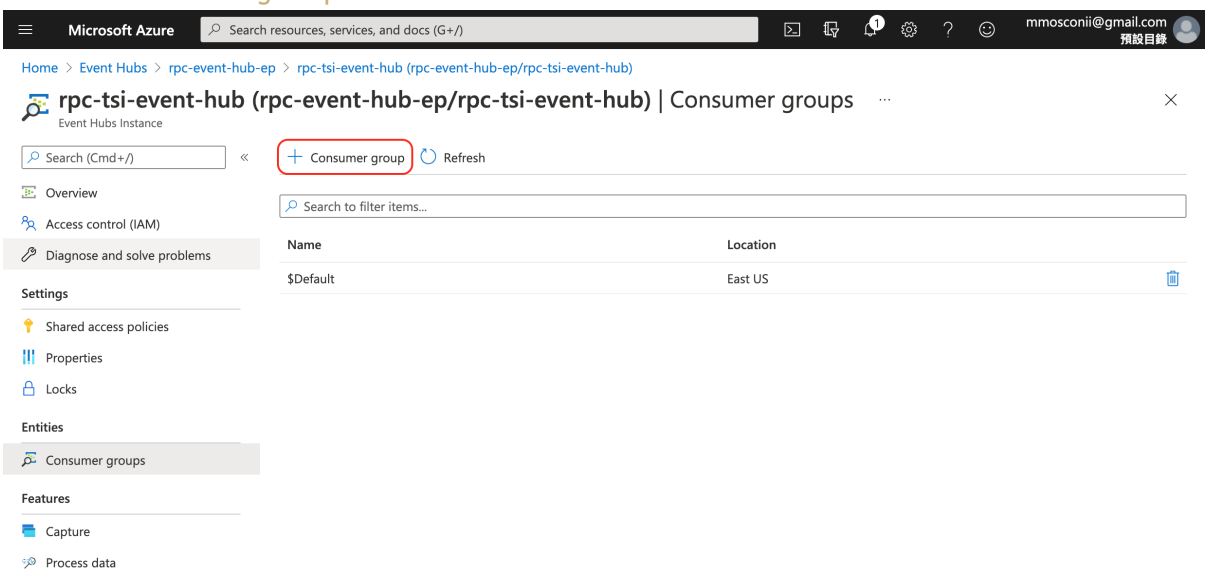
- e.g.

```
az functionapp config appsettings set --settings
"EventHubAppSetting-
TSI=Endpoint=Endpoint=sb://rpchubspace.servicebus.windows.net/;Sh
aredAccessKeyName=rpctsieventrules;SharedAccessKey=xFE9i04ojQHDRQ
kZ4F8J6mIsWoo9o4sAVyVbn/uv7tk=;EntityPath=rptsieventhub" -g rpc-
adt-rg -n rpcEmitAdtTsiFunctions
```

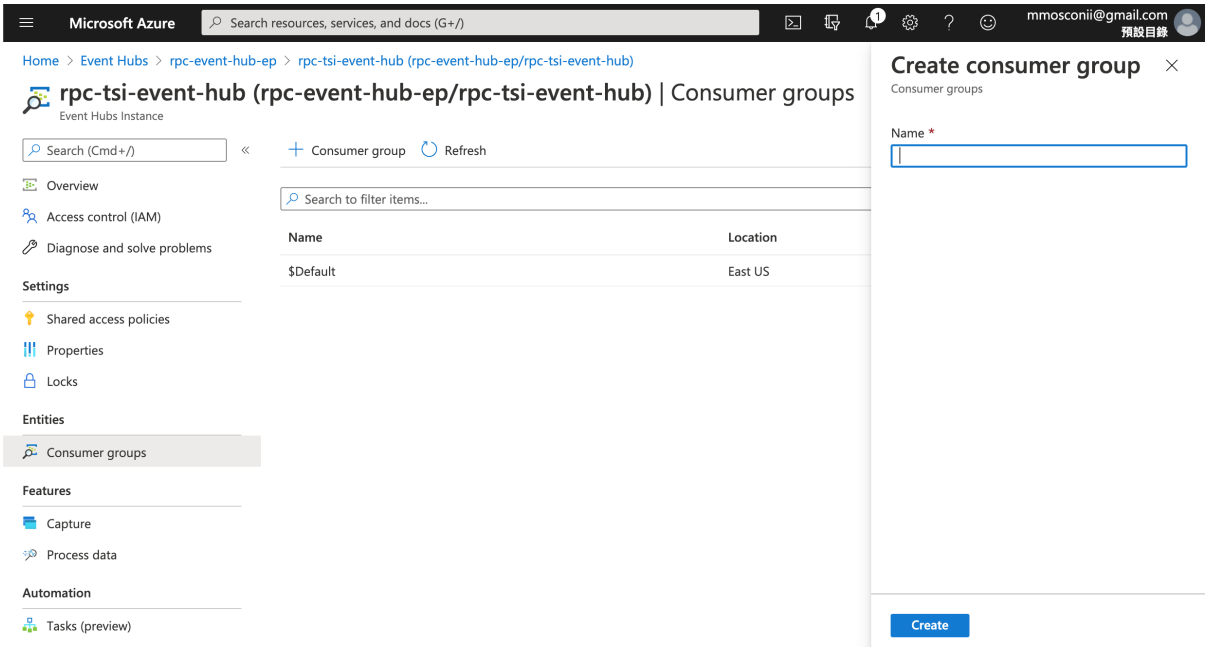
- 9.15 Create `Consumer groups` of rptsieventhub
  - Please select the `Consumer groups` created in `Step 9.7`



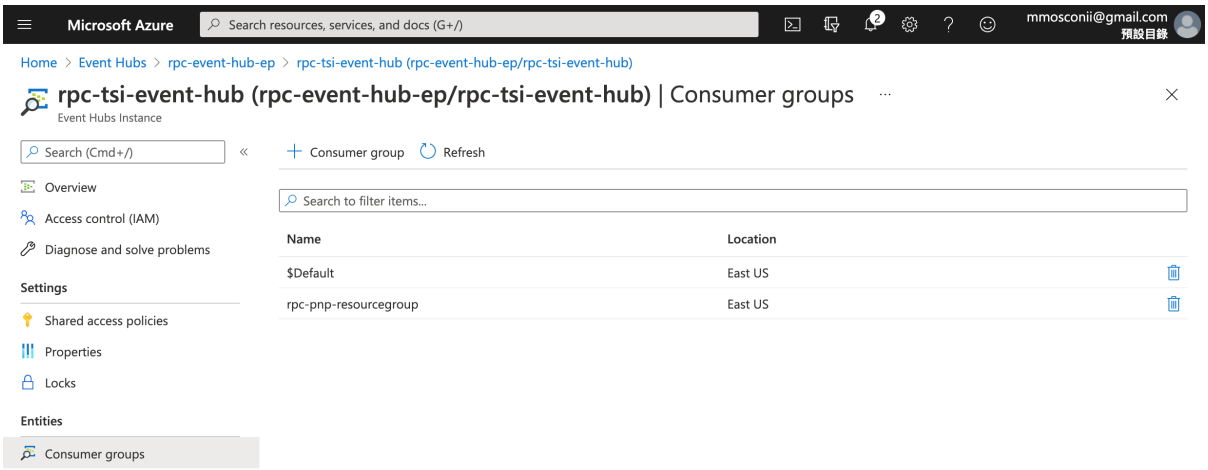
  - Click `+ Consumer groups`



- Input data

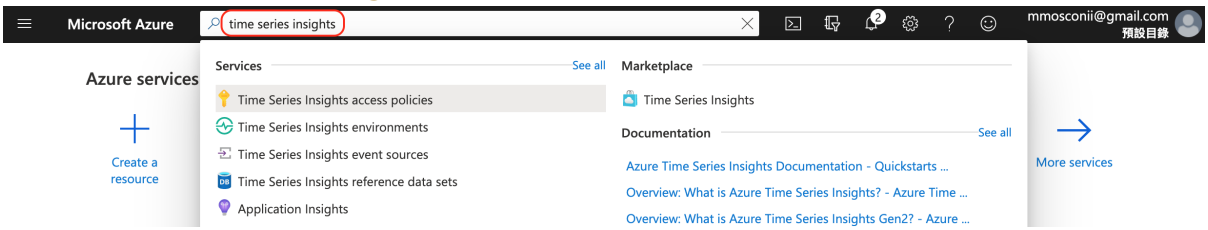


  - Name field
    - Please use a recognizable name, this example uses `rpc-pnp-resourcegroup`
  - When the input is complete, please click the `Create` button
- Operation completes

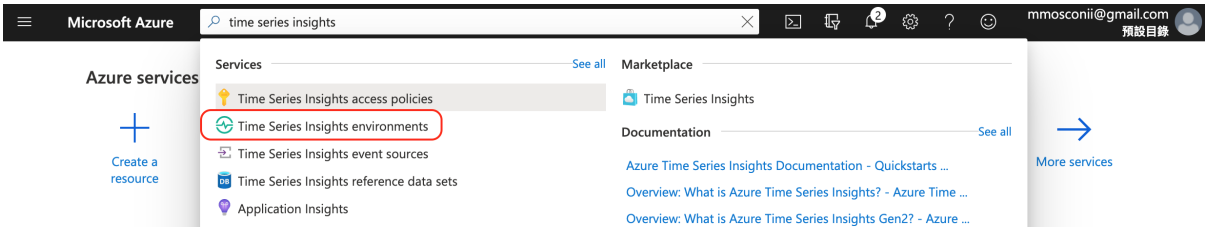


10. Create and set TSI Service

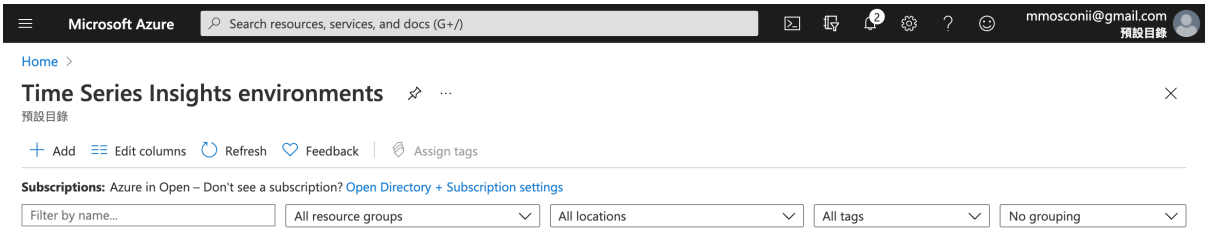
- 10.1 Create and connect a Time Series Insights instance
  - Search `time series insights`



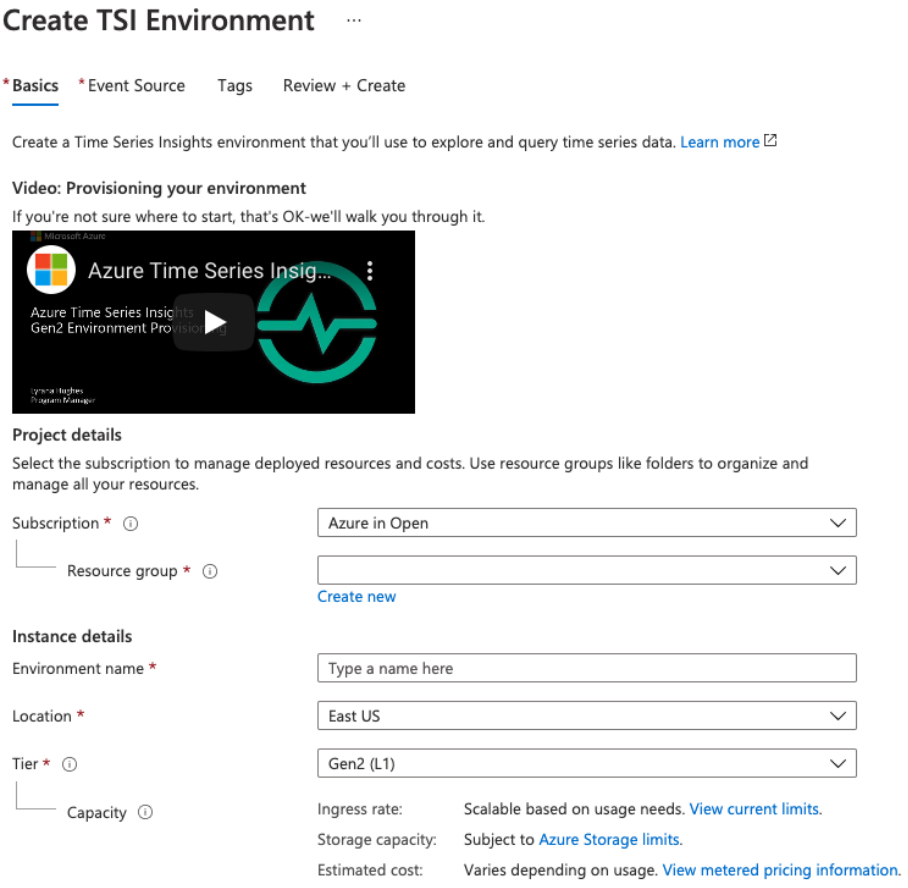
- Select `Time Series Insights environments`

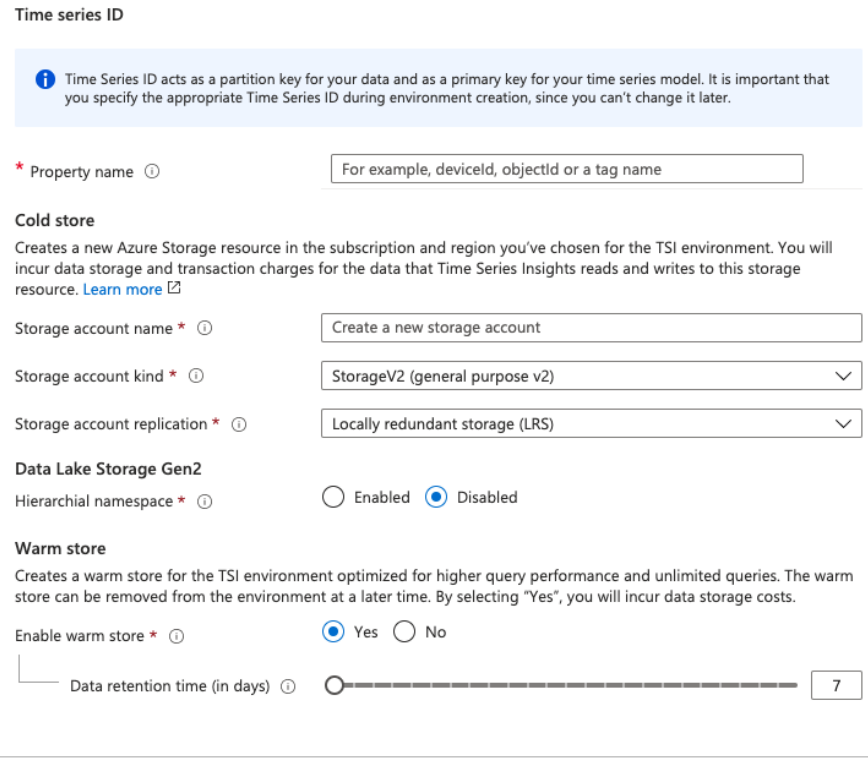


- Click `Add`



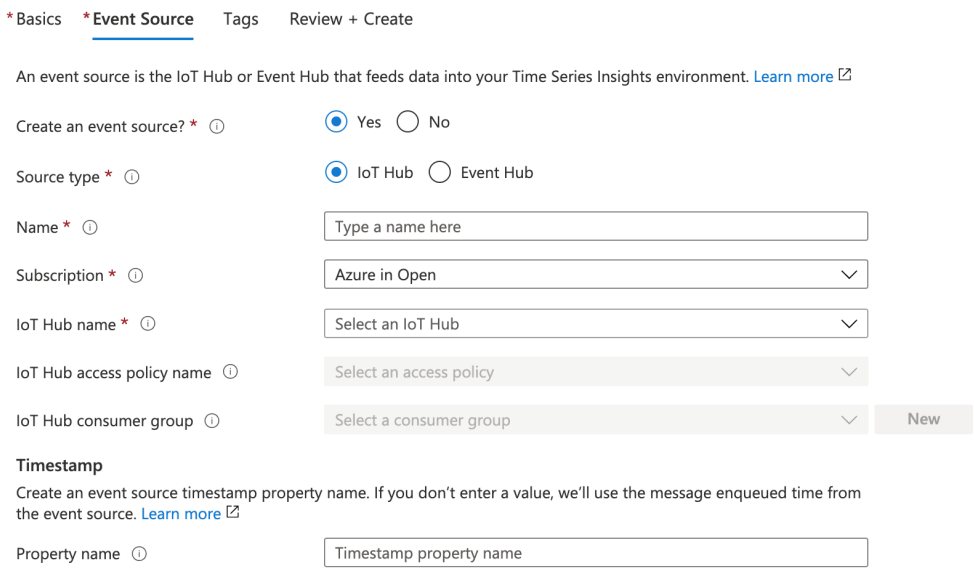
- Input data
  - Basics





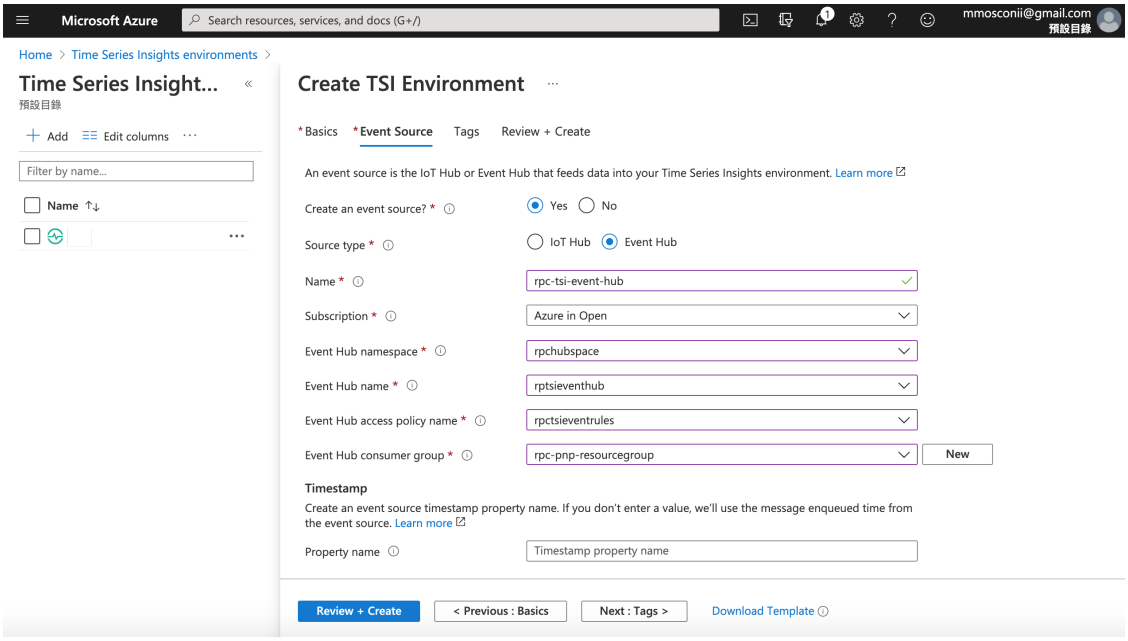
- `Subscription` field
  - Select the subscription you want to use
- `Resource group` field
  - Please use a recognizable name, this example uses `rpc-adt-rg`
- `Environment name` field
  - Please use a recognizable name, this example uses `rpc-tsi`
- `Location` field
  - This example uses `East US`
- `Tier` field
  - This example uses `Gen2 (L1)`
- `Property name` field
  - Please use a recognizable name, this example uses `ADTid`、`Timestamp`
- `Storage account name` field
  - Please use a recognizable name, this example uses `rpctsi`
- `Storage account kind` field
  - This example uses `StorageV2 (general purpose v2)`
- `Storage account replication` field
  - This example uses `Locally redundant storage (LRS)`
- `Hierarchial namespace` field
  - This example uses `Disabled`
- `Enable warm store` field
  - This example uses `No`

- Event Source

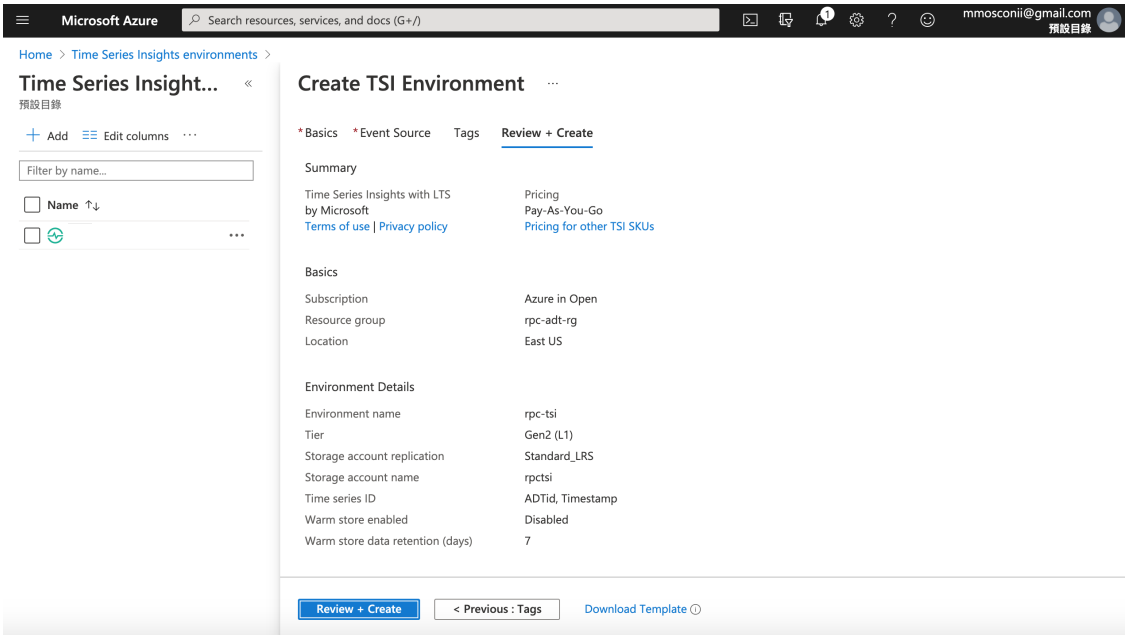


  - `Create an event source?` field
    - This example uses `Yes`
  - `Source type` field
    - This example uses `Event Hub`
  - `Name` field
    - This example uses `rpc-tsi-event-hub`
  - `Subscription` field
    - Select the subscription you want to use
  - `Event Hub namespace` field
    - Please select the `Endpoint` created in `Step 9.1` (rpchubspace)
  - `Event Hub name` field
    - Please select the `TSI Event Hub` created in `Step 9.7` (rptsieventhub)
  - `Event Hub access policy name` field
    - Please select the `Shared access policies` created in `Step 9.8` (rpctsieventrules)
  - `Event Hub consumer group` field
    - Please select the `Consumer groups` created in `Step 9.15` (rpc-pnp-resourcegroup)

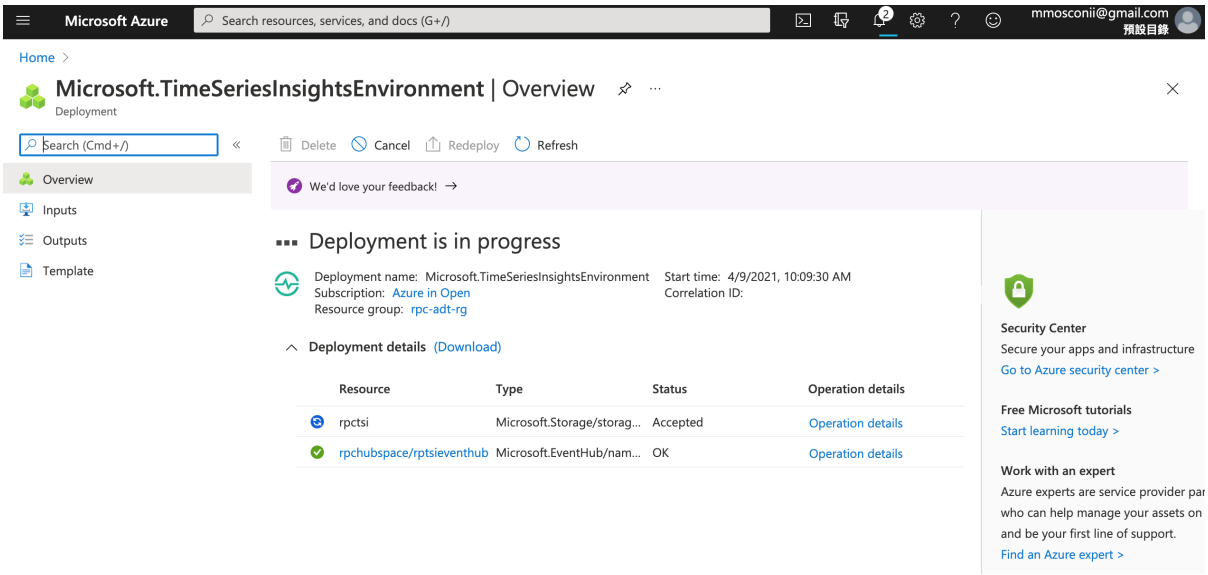
- When the input is complete, please click the `Review + create` button



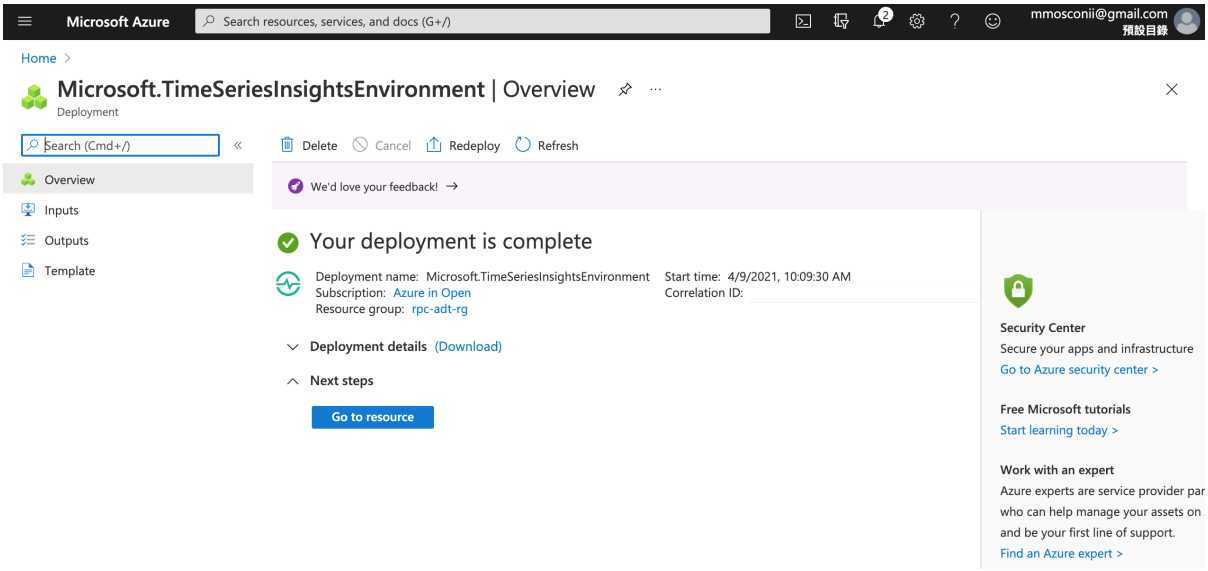
- Review your settings and select `Review + create`



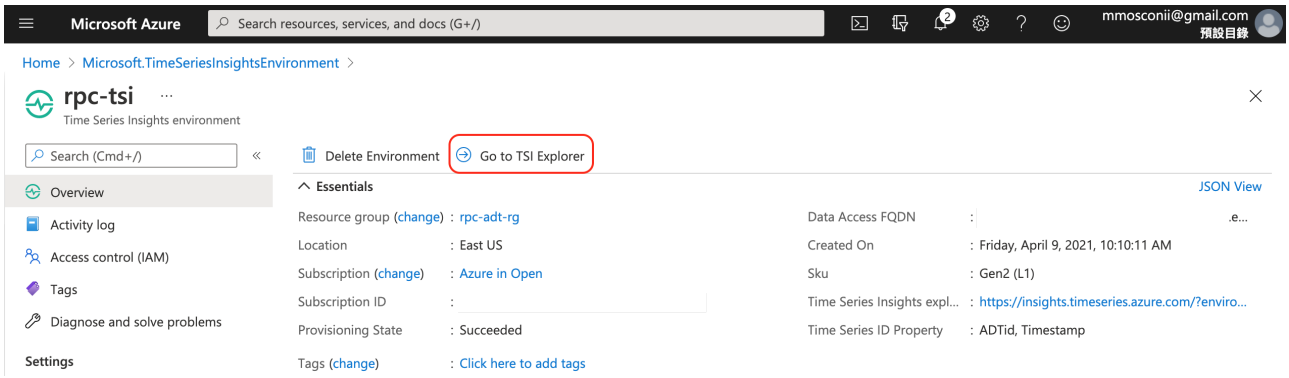
- wait for the request to process



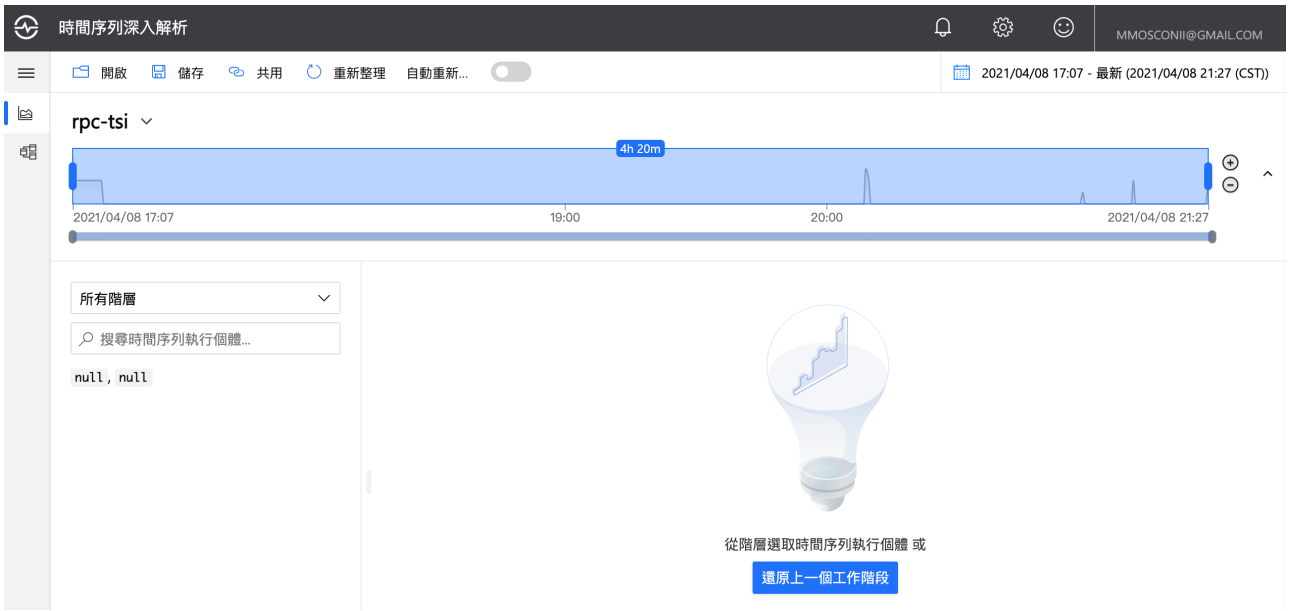
- Once deployment complete click on `Go to resource` button

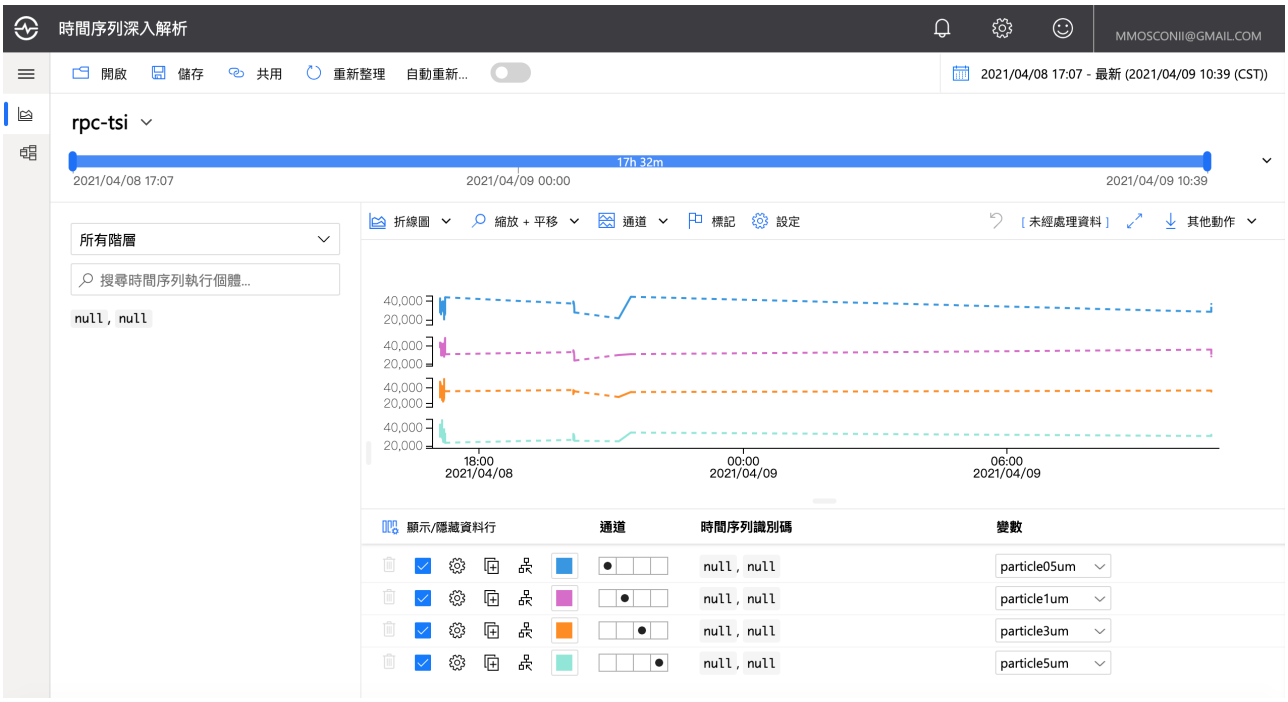


- 10.2 Get to TSI Explorer



- 10.3 TSI Explorer View





11. Set Event & toMap Function

- 11.1 Create a route and filter to twin update notifications
  - Create an event grid topic, which will receive events from your Azure Digital Twins instance.
    - Azure CLI

```
az eventgrid topic create -g <your-resource-group-name> --
name <your-topic-name> -l <region>
```

    - e.g.

```
az eventgrid topic create -g rpc-adt-rg --name rpcmap -l
eastus
```

  - Create an endpoint to link your event grid topic to Azure Digital Twins.
    - Azure CLI

```
az dt endpoint create eventgrid --endpoint-name <Event-Grid-
endpoint-name> --eventgrid-resource-group <Event-Grid-
resource-group-name> --eventgrid-topic <your-Event-Grid-
topic-name> -n <your-Azure-Digital-Twins-instance-name> --
resource-group <resource group name>
```

    - e.g.

```
az dt endpoint create eventgrid --endpoint-name
rpcadteventgridmap --eventgrid-resource-group rpc-adt-rg --
eventgrid-topic rpcmap -n rpc-adt-example --resource-group
rpc-adt-rg
```

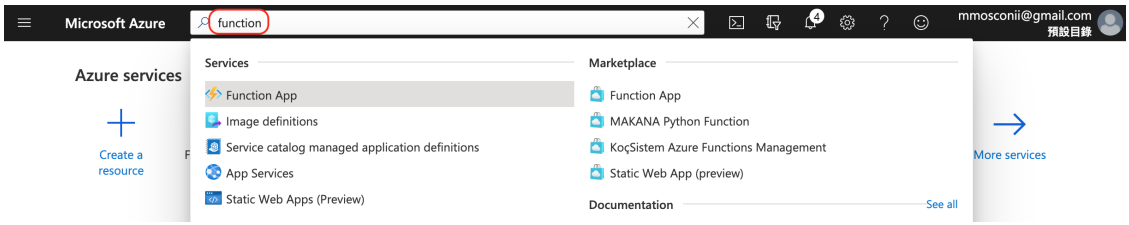
- Create a route in Azure Digital Twins to send twin update events to your endpoint.
  - Azure CLI

```
az dt route create -n <your-Azure-Digital-Twins-instance-
name> --endpoint-name <Event-Grid-endpoint-name> --route-
name <my_route> --filter "type =
'Microsoft.DigitalTwins.Twin.Update'" --resource-group
<resource group name>
```

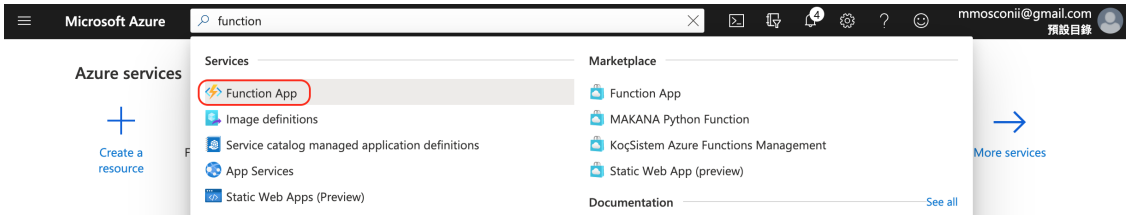
  - e.g.

```
az dt route create -n rpc-adt-example --endpoint-name
rpcadteventgridmap --route-name rpcadtmaproute --filter
"type = 'Microsoft.DigitalTwins.Twin.Update'" --resource-
group rpc-adt-rg
```

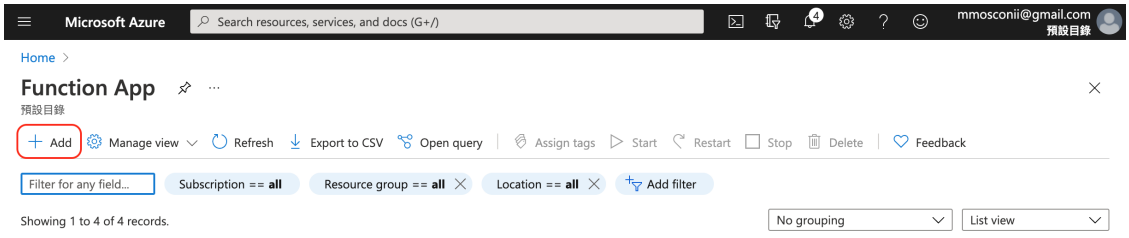
- 11.2 Create & Deploy a function in Azure
  - 11.2.1 Create Azure Functions
    - Search `function`



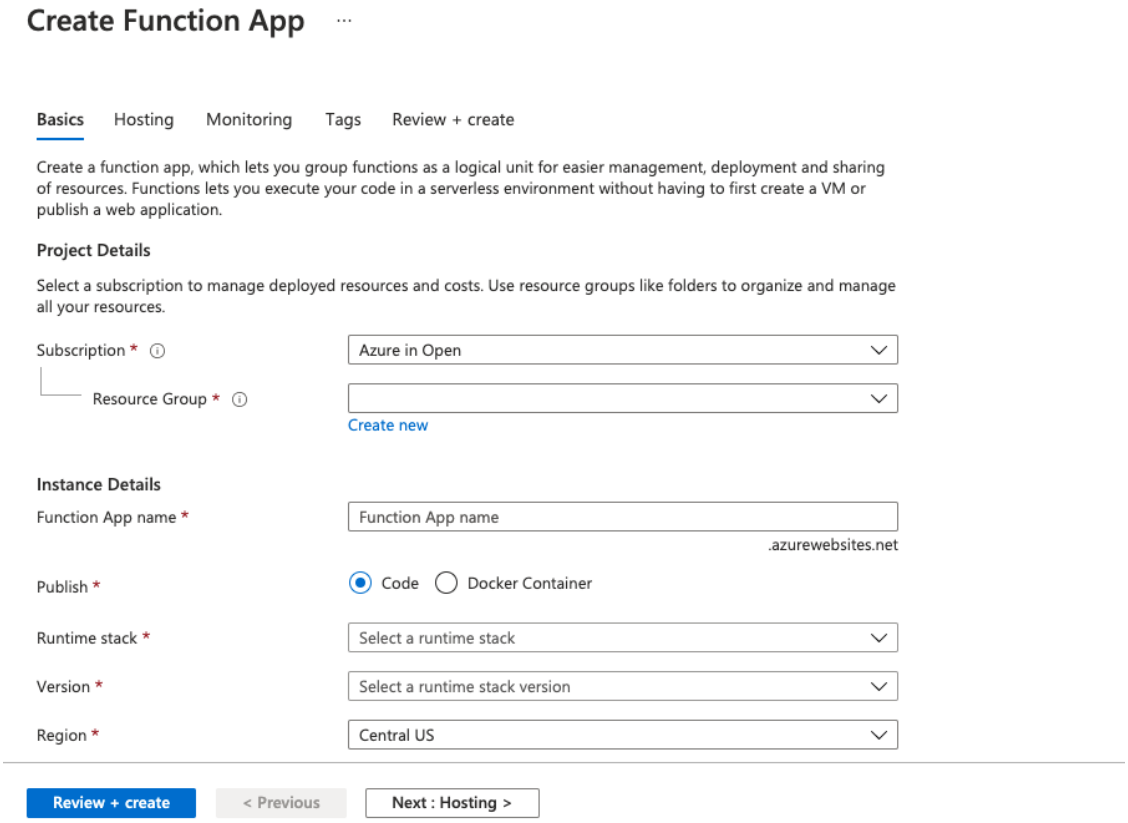
    - Select `Function App`



    - Click `Add`



- Input data



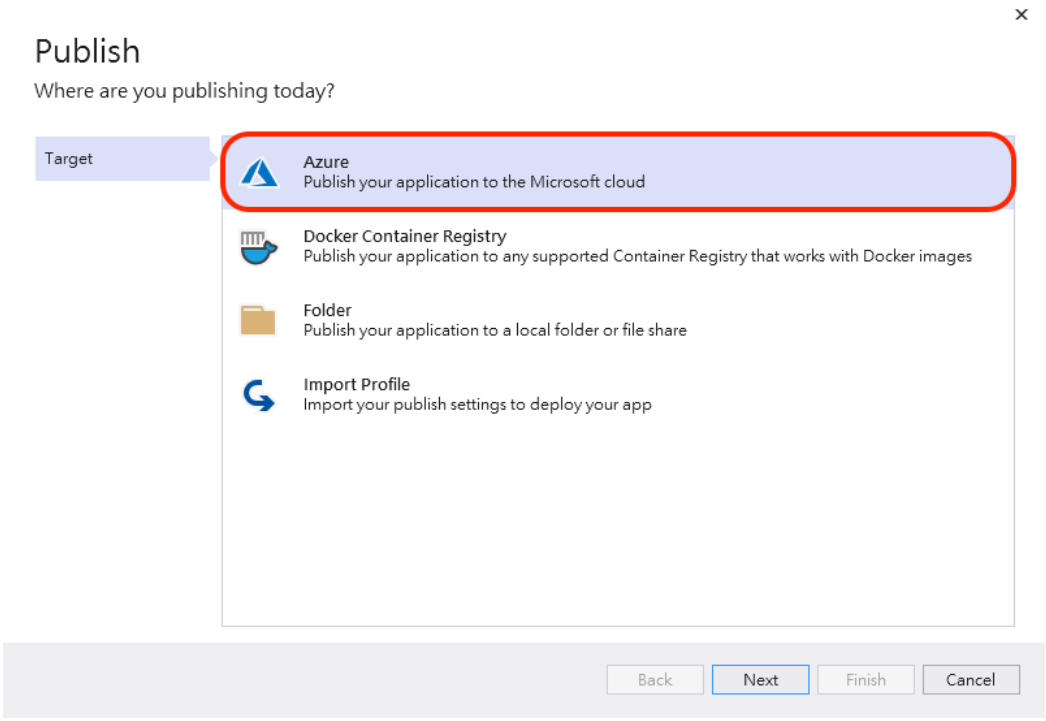
  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please use a recognizable name
  - `Function App name` field
    - Please use a recognizable name, this example uses `rpcToMapFunctions`
  - `Publish` field
    - Select `Code`
  - `Runtime stack` field
    - Select `.NET`
  - `Version` field
    - Select `3.1`
  - `Region` field
    - This example uses `East US`
  - Click on the `Review + create` button
  - Review all the details and click on the `Create` button
  - wait for the request to process
  - 建立完成，請點選 `Go to resource`

- 11.2.2 Deploy

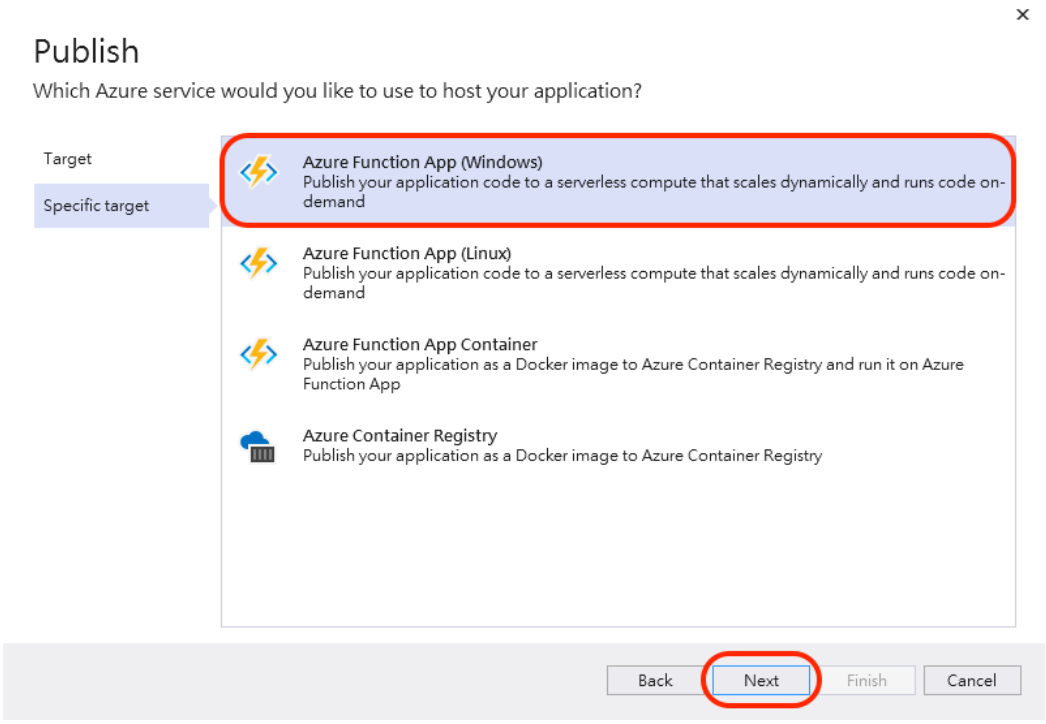
  - Opening `toMap.sln` from `/Back-End/Azure-Functions/toMap` in `Visual Studio 2019`
  - Deploy `toMap` to `Azure Function`

    - Click `Publish...`

- Click `+ New`
- Click `Azure`



- Click `Azure Function App (Windows)` > `Next`



- Please select the function created in `Step 11.2.1`
- Click `Publish`
- Operation completes

- 11.2.3 Set Configuration
  - Azure Maps primary subscription key

- Azure CLI

```
az functionapp config appsettings set --name <your-App-Service-(function-app)-name> --resource-group <your-resource-group> --settings "subscription-key=<your-Azure-Maps-primary-subscription-key>"
```

- e.g.

```
az functionapp config appsettings set --name rpcToMapFunctions --resource-group rpc-adt-rg --settings "subscription-key=yEM2lU7LOKbjNJfJ5AujcoOscWRjQZ-tyG5g4pudf4o"
```

- Azure Maps stateset ID

- Azure CLI

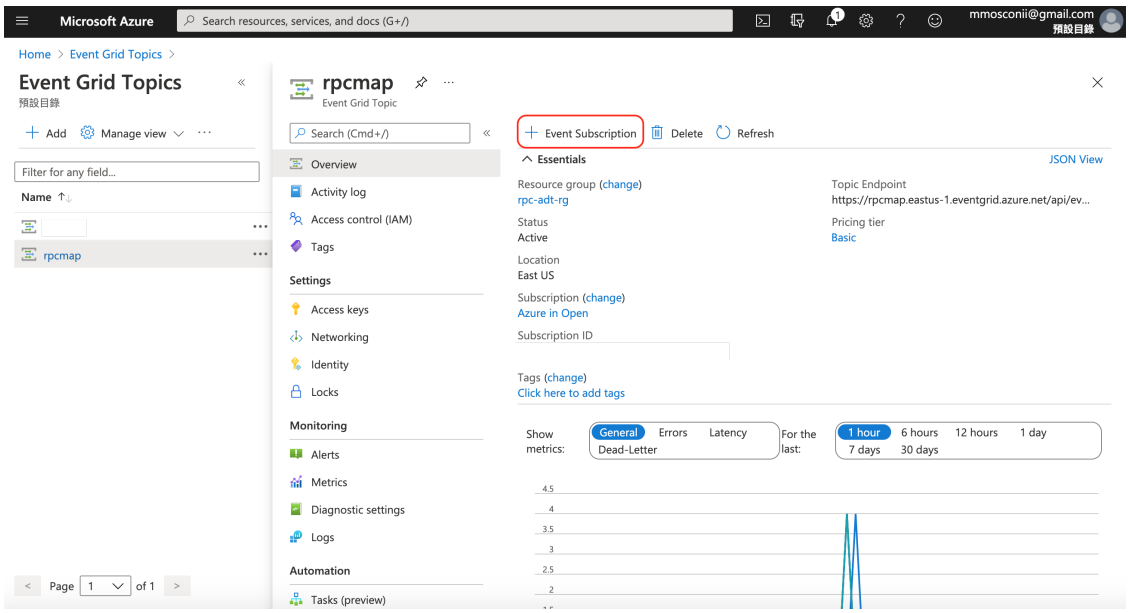
```
az functionapp config appsettings set --name <your-App-Service-(function-app)-name>  --resource-group <your-resource-group> --settings "statesetID=<your-Azure-Maps-stateset-ID>"
```

- e.g.

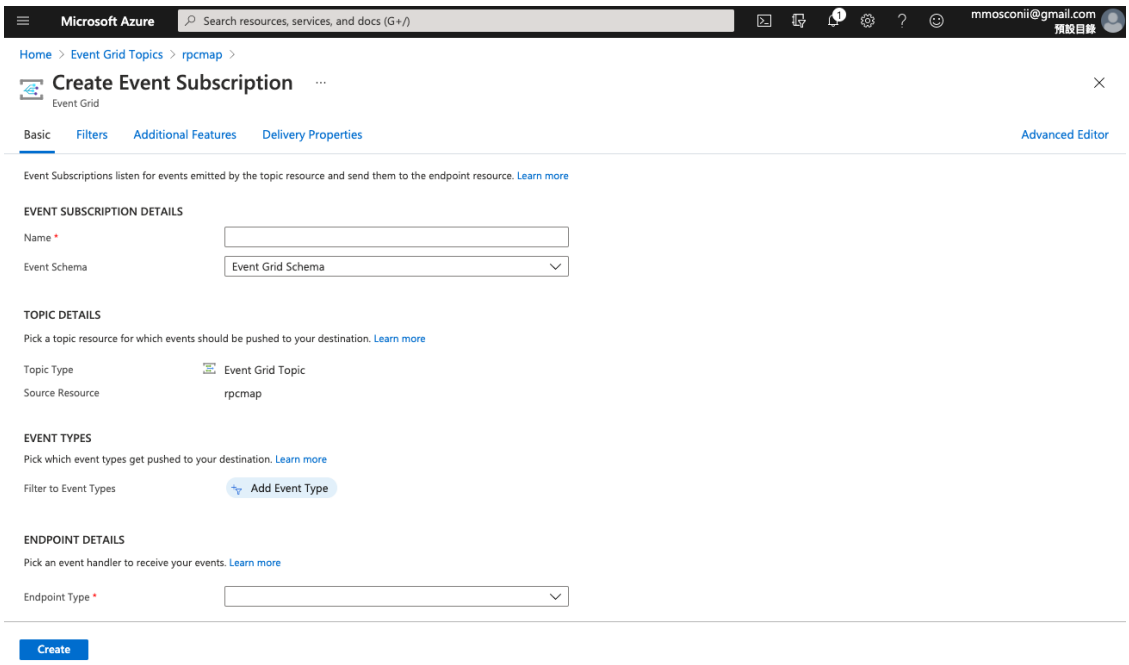
```
az functionapp config appsettings set --name rpcToMapFunctions --resource-group rpc-adt-rg --settings "statesetID=0dac1372-a147-b3b3-9d93-6765981b9213"
```

- 11.2.4 Set Event Subscription

- Click `+ Event Subscription`

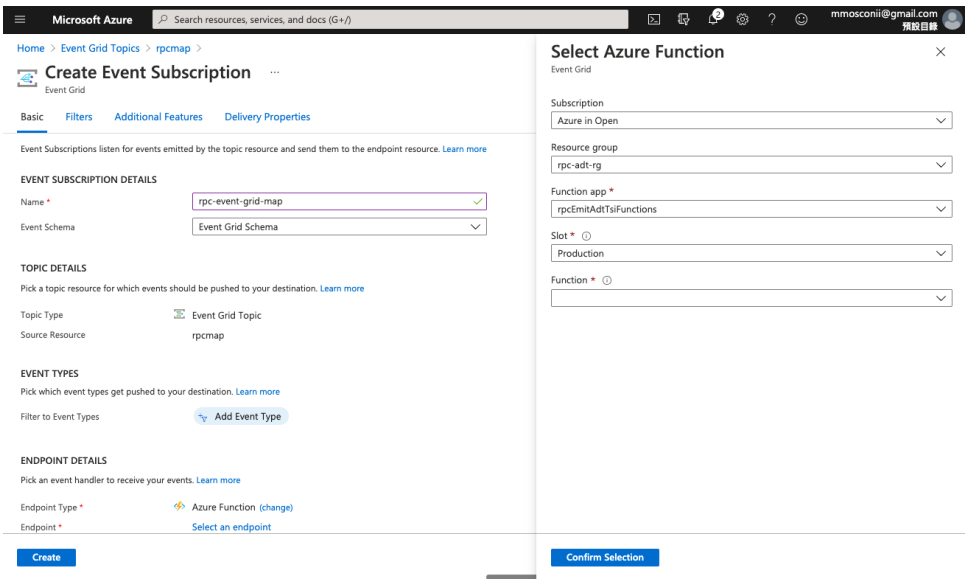


- Input data

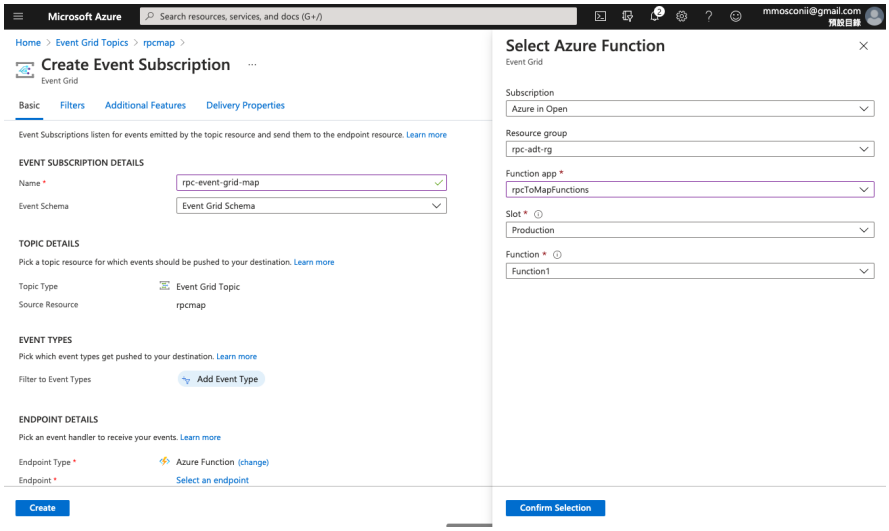


  - `Name` field
    - Please use a recognizable name, this example uses `rpc-event-grid-map`
  - `Endpoint Type` field
    - Select `Azure Function`
  - `Endpoint` field
    - Click `Select an endpoint`

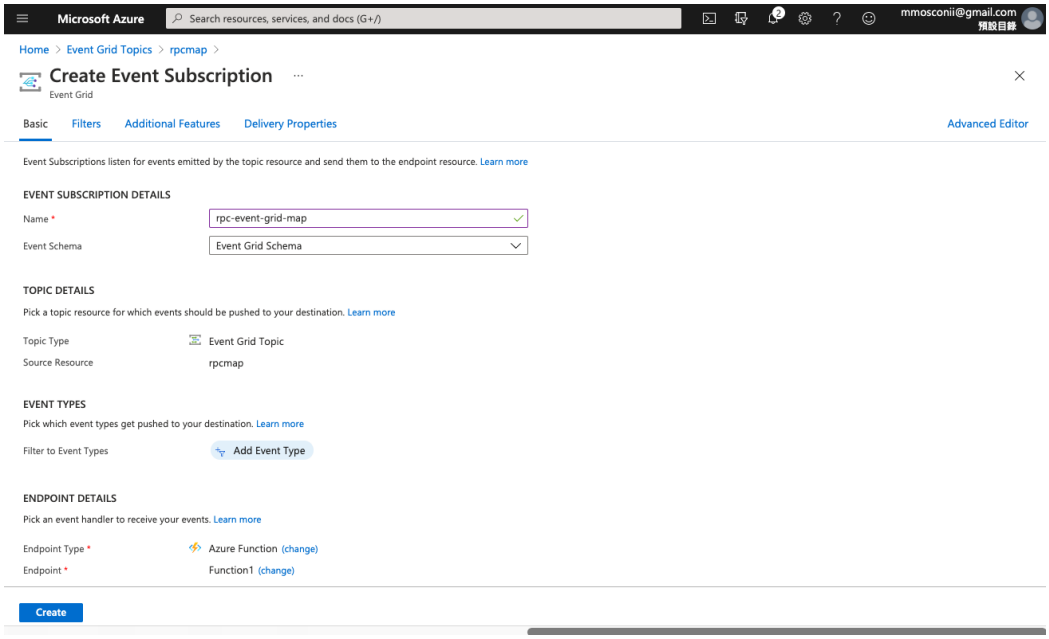
- `Select Azure Function` view



  - `Subscription` field
    - Select the subscription you want to use
  - `Resource group` field
    - Please select the `resource group` created in `Step 3`
  - `Function app` field
    - Please select the function created in `Step 11.2`
  - `Slot` field
    - Select `Production`
  - `Function` field
    - Select `Function1`
  - When the input is complete, please click the `Confirm Selection` button

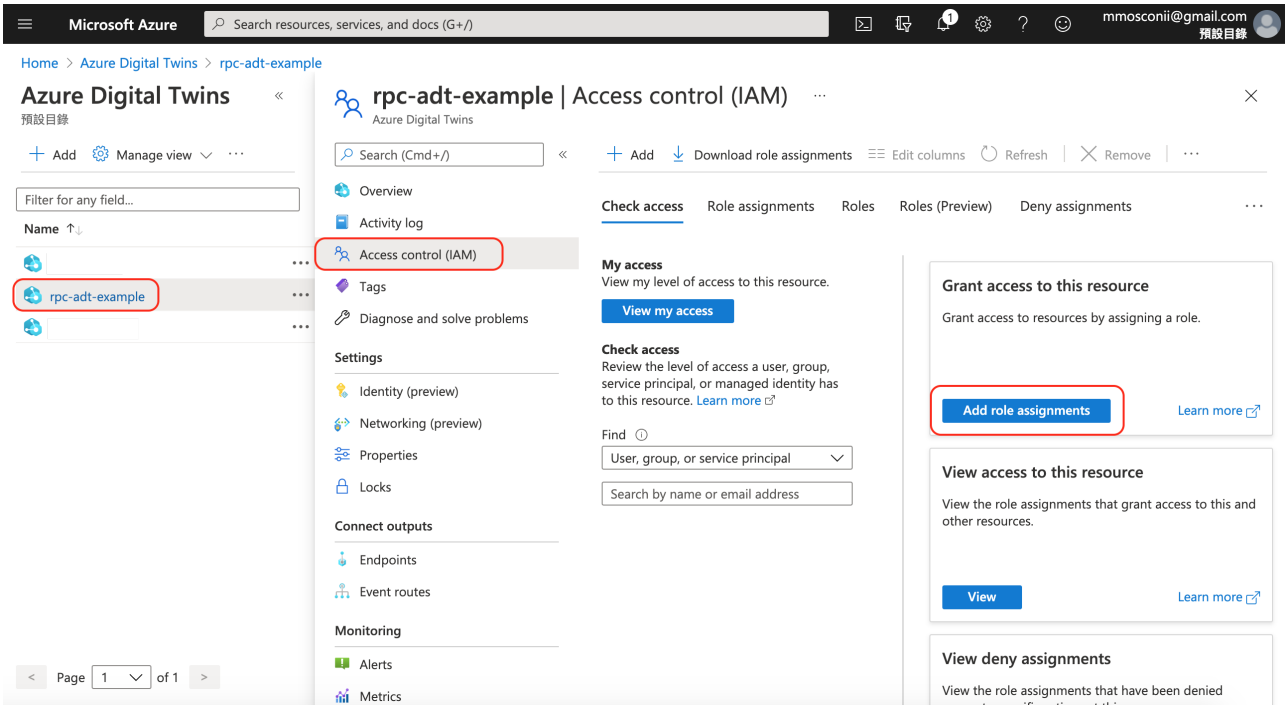


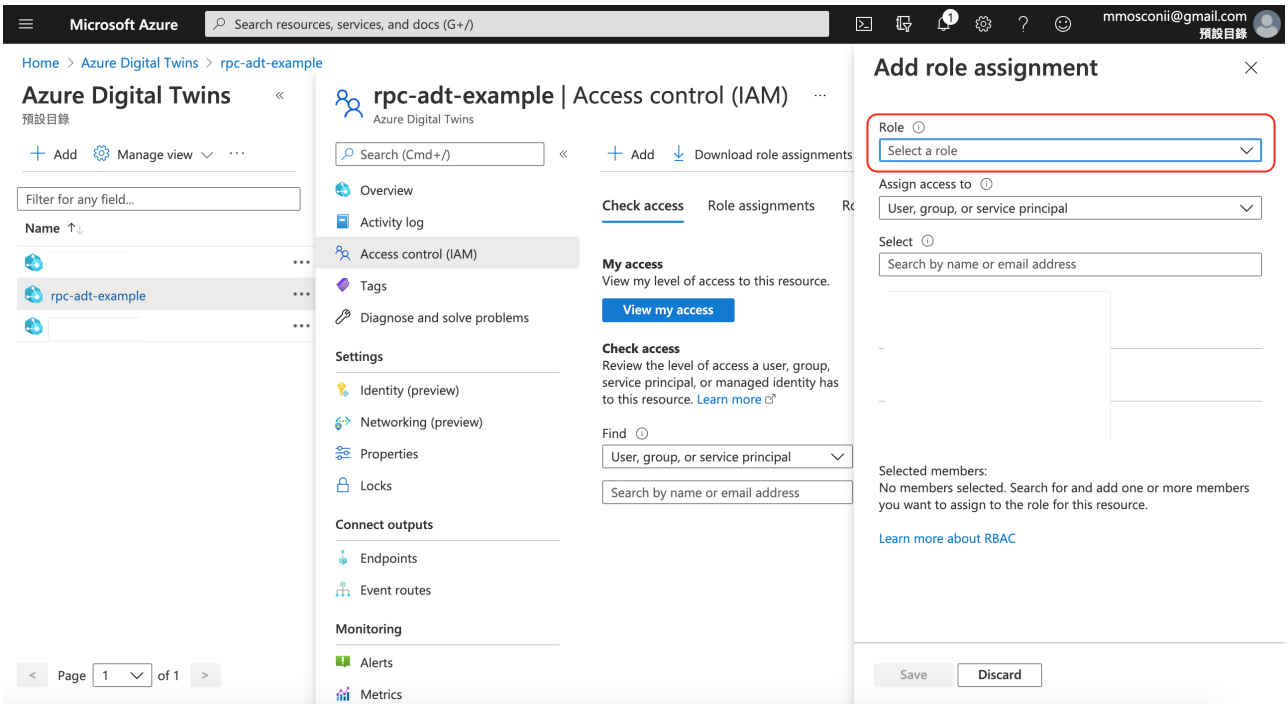
- When the input is complete, please click the `Create` button

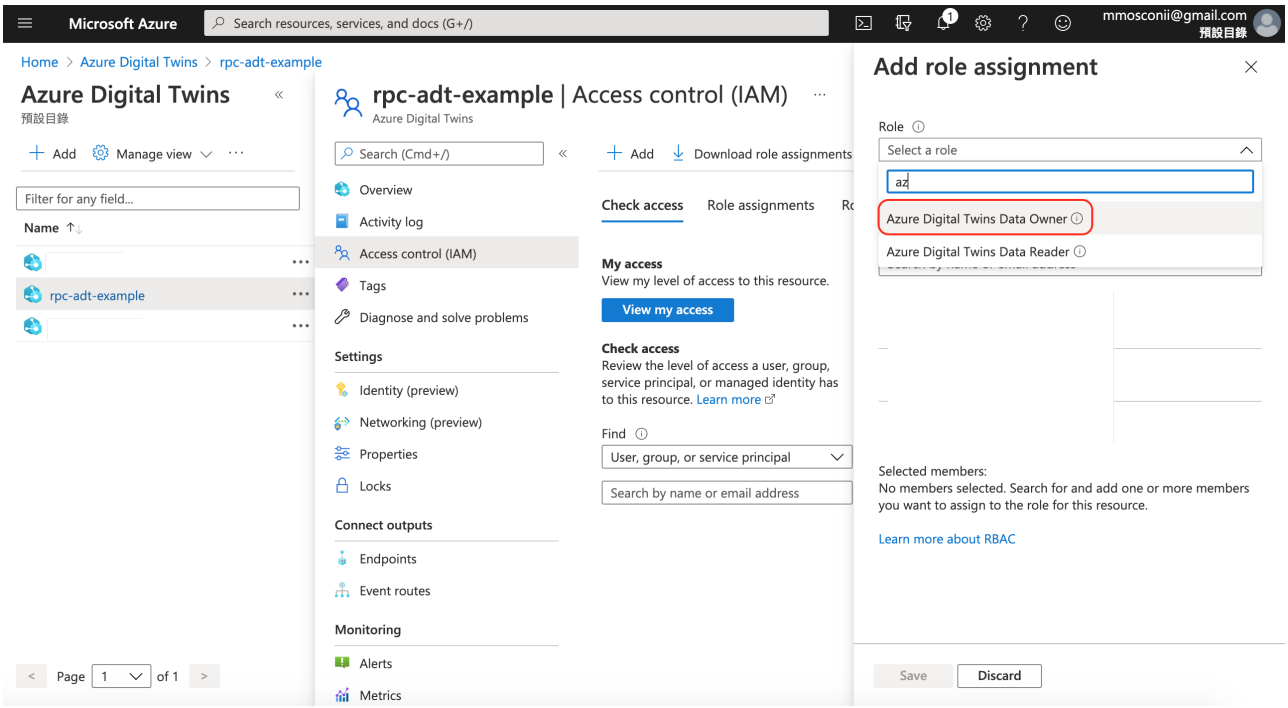


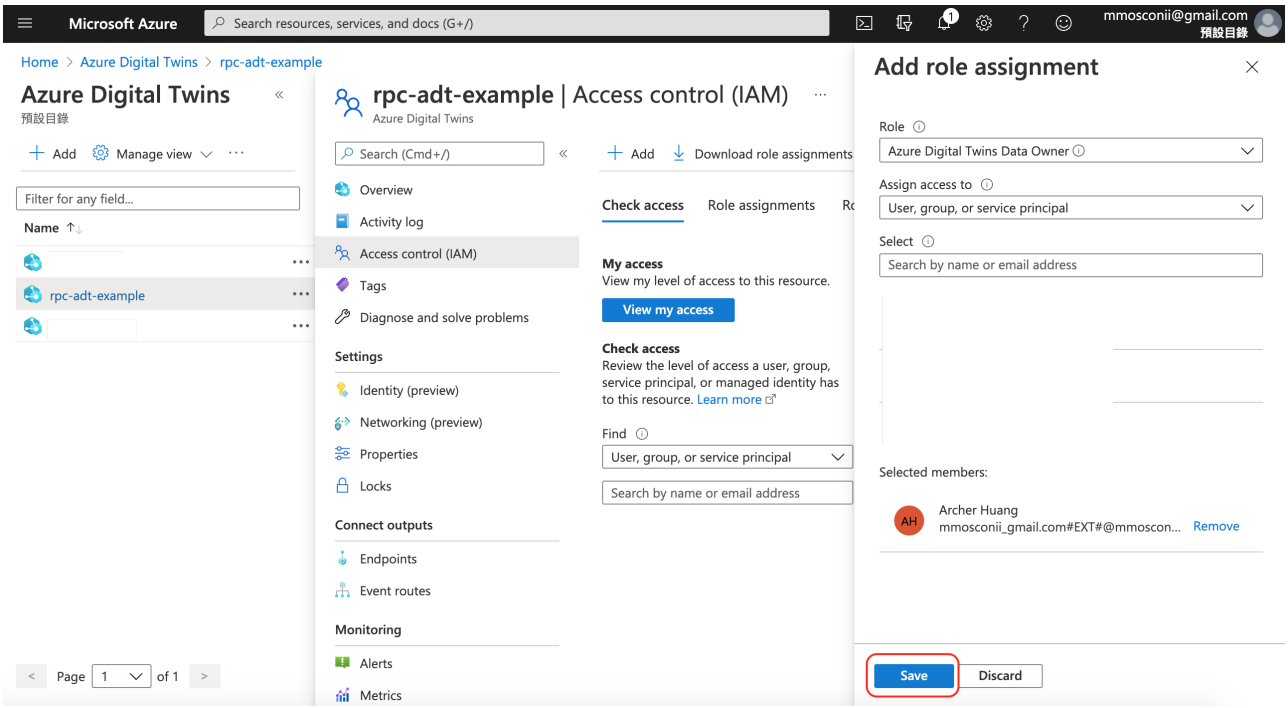
12. Set Dashboard and Upload Model

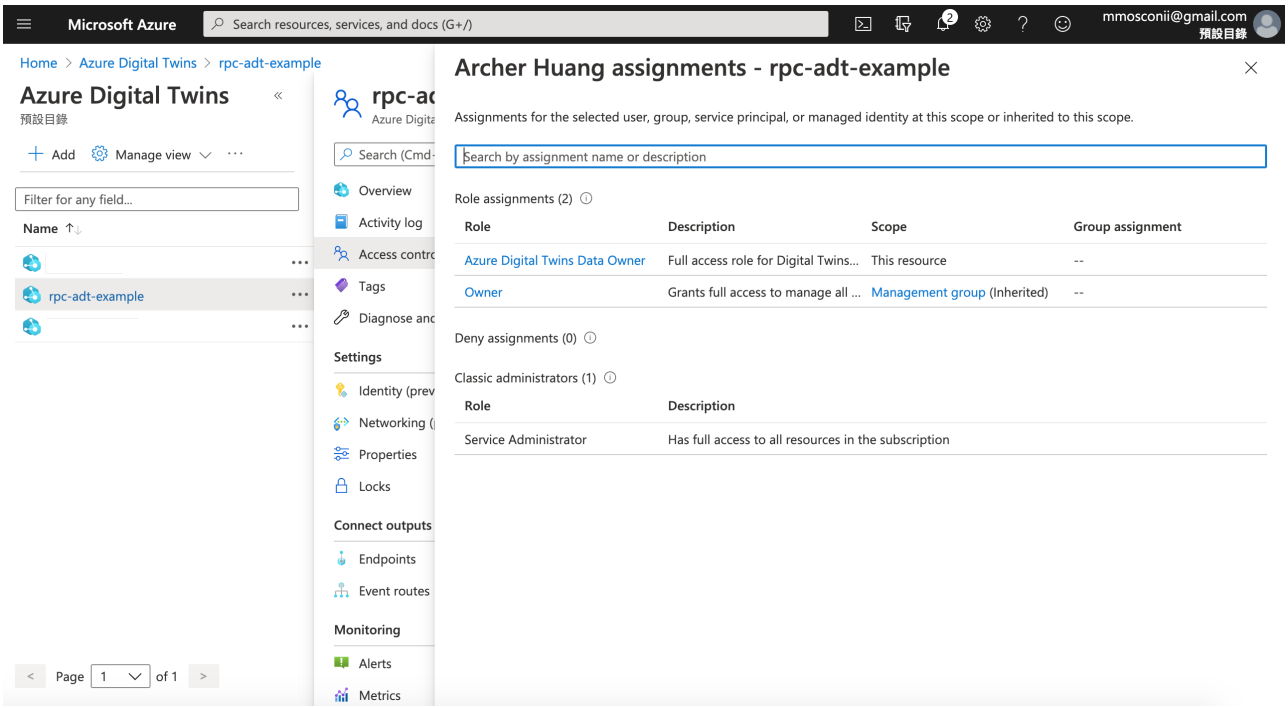
- 12.1 Set role assignment for the Azure Digital Twins




Microsoft Azure
Search resources, services, and docs (G+/)
mmosconii@gmail.com
預設目錄
Home > Azure Digital Twins > rpc-adt-example
Azure Digital Twins
預設目錄
Add
Manage view
Filter for any field...
Name
rpc-adt-example
Page 1 of 1
rpc-adt-example | Access control (IAM)
Azure Digital Twins
Search (Cmd+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Identity (preview)
Networking (preview)
Properties
Locks
Connect outputs
Endpoints
Event routes
Monitoring
Alerts
Metrics
Add
Download role assignments
Check access
Role assignments
My access
View my access
Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more
Find
User, group, or service principal
Search by name or email address
Add role assignment
Role
Select a role
Assign access to
User, group, or service principal
Select
Search by name or email address
Selected members:
No members selected. Search for and add one or more members you want to assign to the role for this resource.
Learn more about RBAC
Save
Discard


Microsoft Azure
Search resources, services, and docs (G+/)
mmosconii@gmail.com
預設目錄
Home > Azure Digital Twins > rpc-adt-example
Azure Digital Twins
預設目錄
Add
Manage view
Filter for any field...
Name
rpc-adt-example
Page 1 of 1
rpc-adt-example | Access control (IAM)
Azure Digital Twins
Search (Cmd+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Identity (preview)
Networking (preview)
Properties
Locks
Connect outputs
Endpoints
Event routes
Monitoring
Alerts
Metrics
Add
Download role assignments
Check access
Role assignments
My access
View my access
Check access
Review the level of access a user, group, service principal, or managed identity has to this resource. Learn more
Find
User, group, or service principal
Search by name or email address
Add role assignment
Role
Select a role
az
Azure Digital Twins Data Owner
Azure Digital Twins Data Reader
Selected members:
No members selected. Search for and add one or more members you want to assign to the role for this resource.
Learn more about RBAC
Save
Discard





- 12.2 Install Node.js, npm and Vue on the local server
  - npm
  - Node.js v10.16.0
  - vue v2.6.12
- 12.3 Proxy Server
  - Enter the project folder

```
$ cd ./Azure-Digital-Twins-for-RPC/Back-End/Proxy
```

  - Install packages via npm

```
$ npm install
```

- Create .env file

```
$ touch .env
```

- Store Key in .env file and save
  - Add ./Azure-Digital-Twins-for-RPC/Back-End/Proxy/.env content

```
AZURE_DIGITAL_TWINS_HOST_NAME=https://
```

  - Comment
    - Copy the Host name that was retrieved in Step 3.8
- Activate the server

```
$ node index.js
```

- Find the message for successful activation

```
[HPM] Proxy created: /  -> https://rpc-adt-
example.api.eus.digitaltwins.azure.net
Example app listening on port 3000!
```

- 12.4 Dashboard Server
  - Enter the project folder

```
$ cd ./Azure-Digital-Twins-for-RPC/Front-End
```

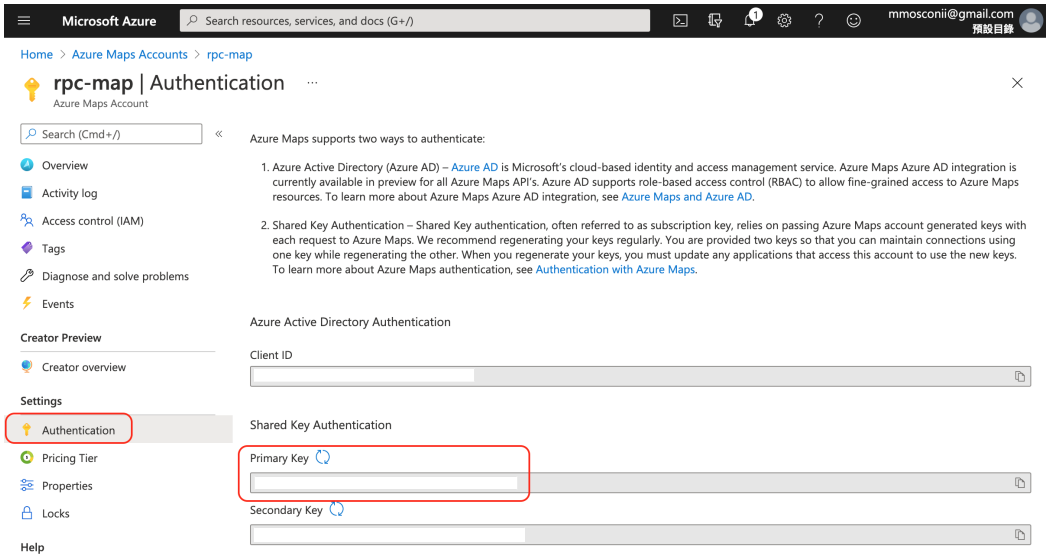
  - Install packages via npm

```
$ npm install
```

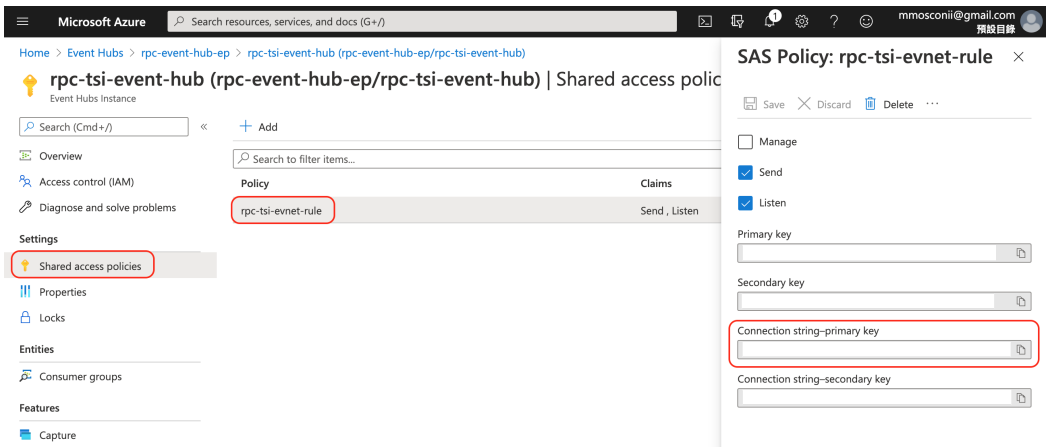
- Create .env file
  - Add ./Azure-Digital-Twins-for-RPC/Front-End/.env content

```
VUE_APP_MAP_SUBSCRIPTION_KEY=
VUE_APP_MAP_TILESETID=
VUE_APP_MAP_STATESETID=
VUE_APP_LONGITUDE=
VUE_APP_LATITUDE=
VUE_APP_COMPANY_NAME=
VUE_APP_EVENT_HUB_CONNECTIONSTRING=
VUE_APP_EVENT_HUB_NAME=
VUE_APP_IOT_HUB_ENDPOINT=
VUE_APP_IOT_HUB_DEVICEKEY=
VUE_APP_IOT_HUB_POLICYNAME=
```

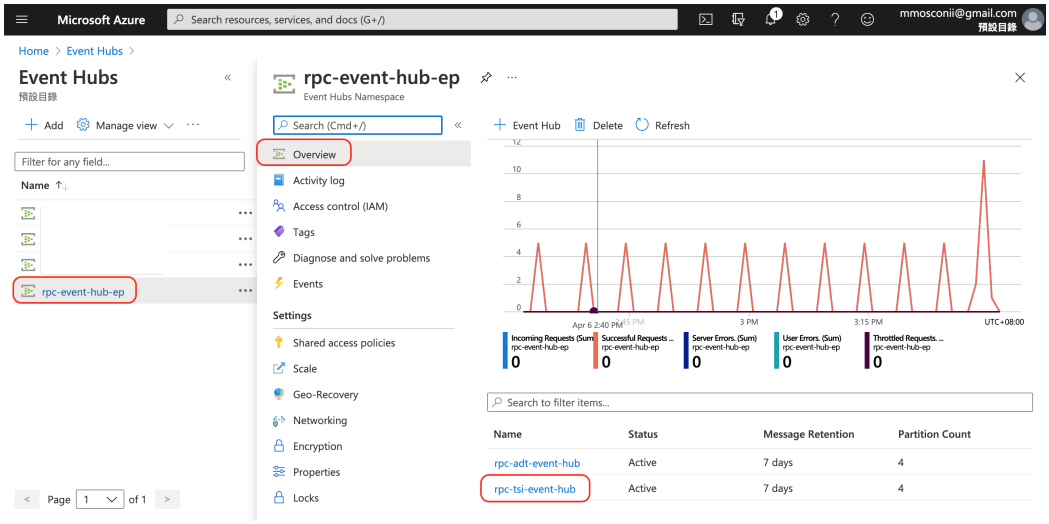
  - Comment
    - VUE_APP_MAP_SUBSCRIPTION_KEY



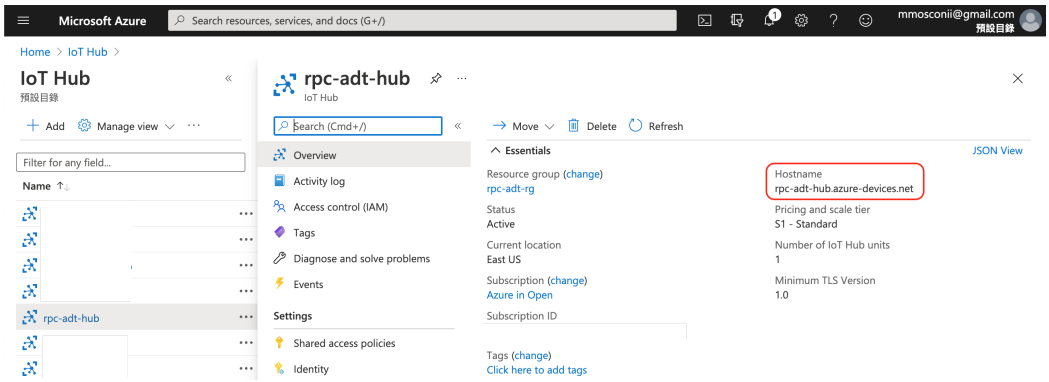
    - VUE_APP_MAP_TILESETID was retrieved in Step 6.8.4
    - VUE_APP_MAP_STATESETID was retrieved in Step 6.11.3
    - VUE_APP_EVENT_HUB_CONNECTIONSTRING



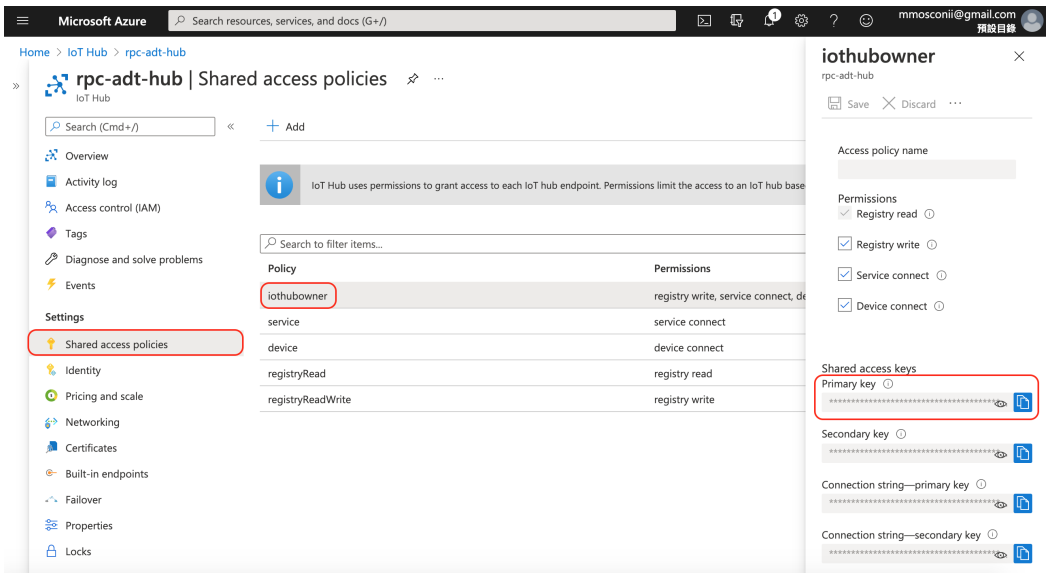
- VUE_APP_EVENT_HUB_NAME



- VUE_APP_IOT_HUB_ENDPOINT



- VUE_APP_IOT_HUB_DEVICEKEY & VUE_APP_IOT_HUB_POLICYNAME

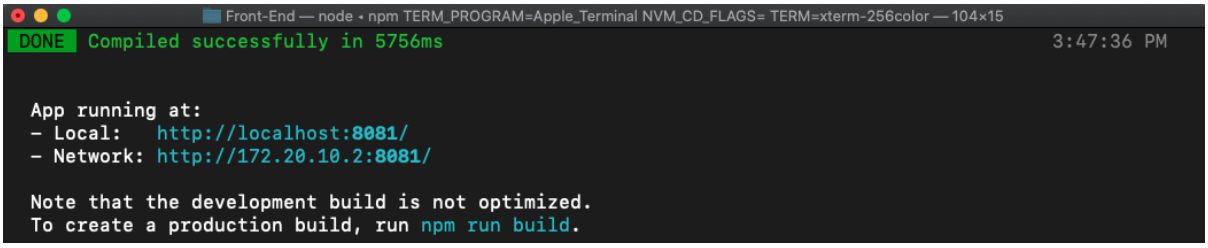


- Activate the server

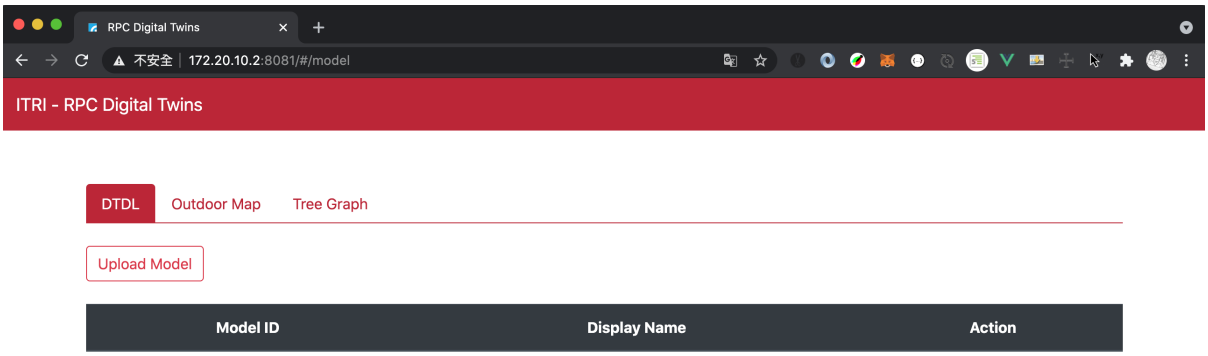
```
$ npm run serve
```

- Find the message for successful activation

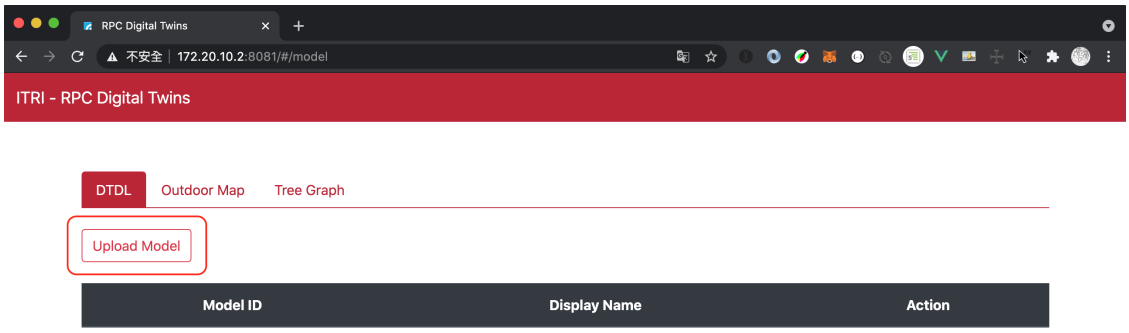
```
App running at:
- Local:   http://localhost:8081/
- Network: http://172.20.10.2:8081/

Note that the development build is not optimized.
To create a production build, run npm run build.
```



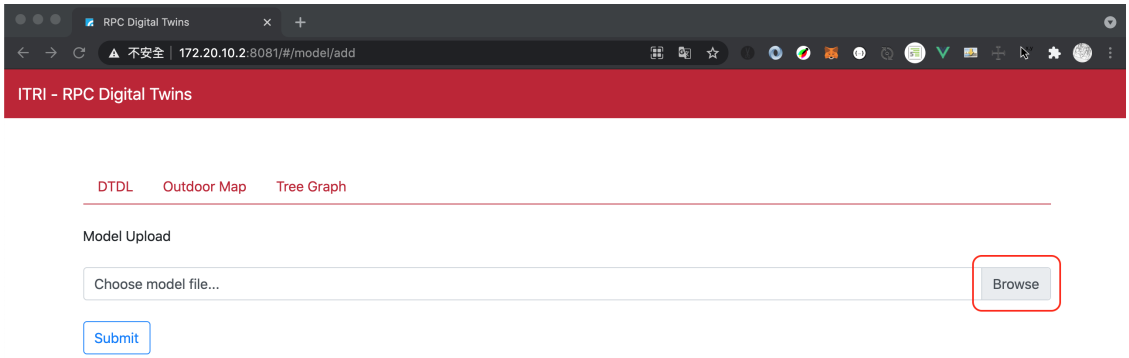
- 12.5 Upload DTDL
  - Open Dashboard



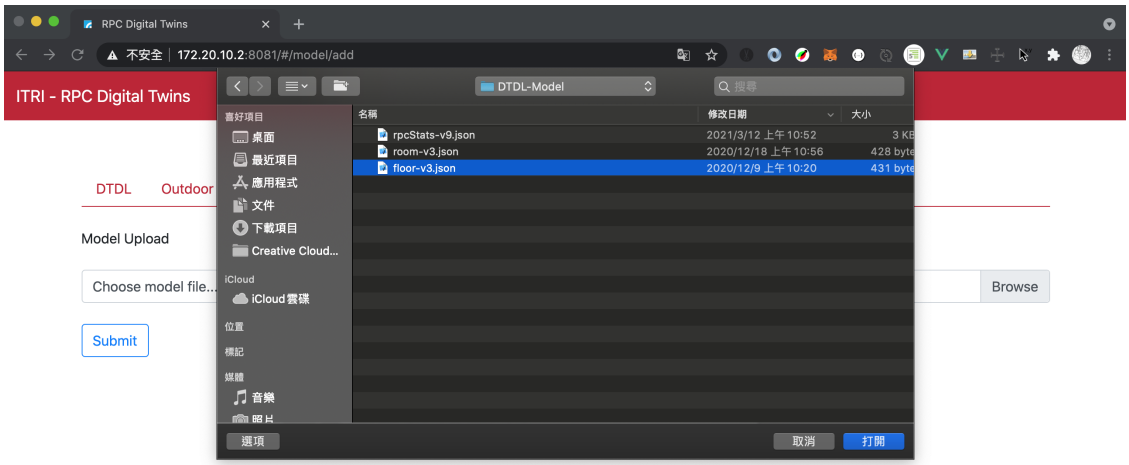
  - Upload floor-v3.json
    - Click `Upload Model`



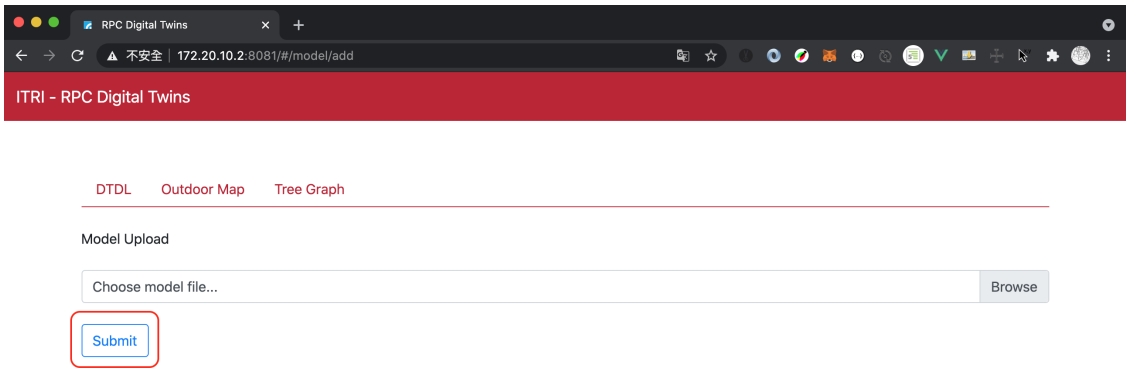
- Click `Browse`



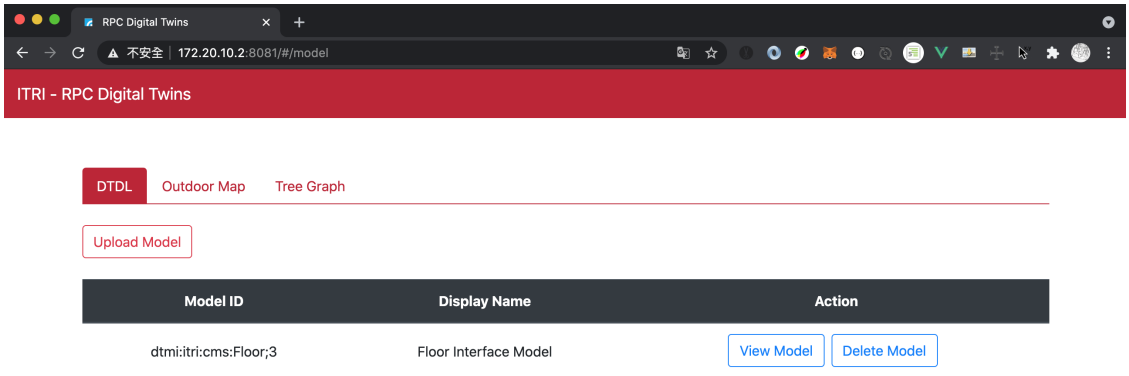
- Select `./Azure-Digital-Twins-for-RPC/DTDL-Model/floor-v3.json`



- Click `Submit`

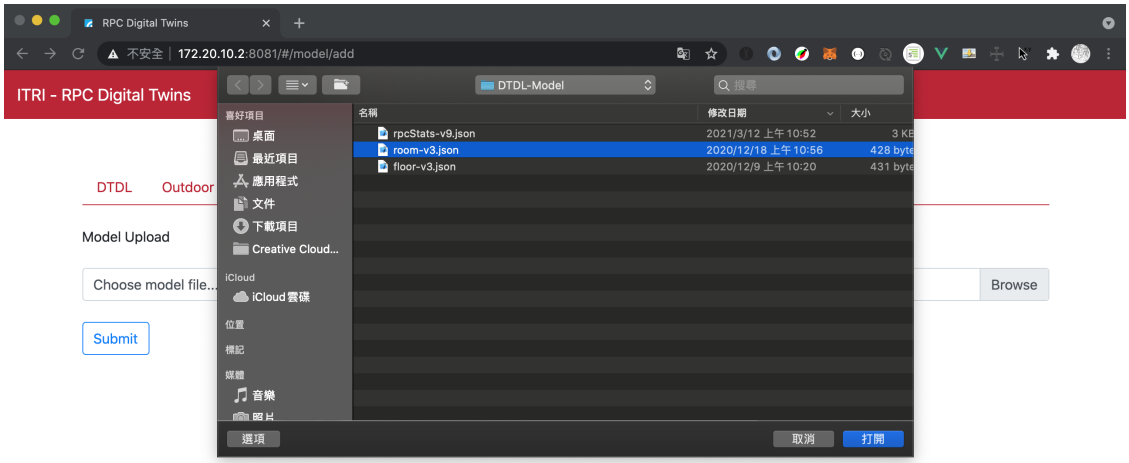


- Upload finish

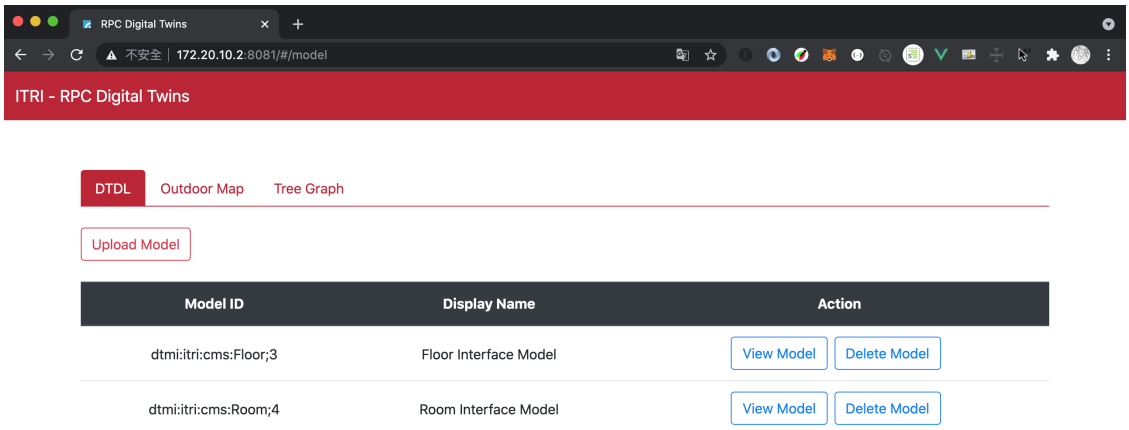


- Upload room-v3.json
  - Click `Upload Model`
  - Click `Browse`

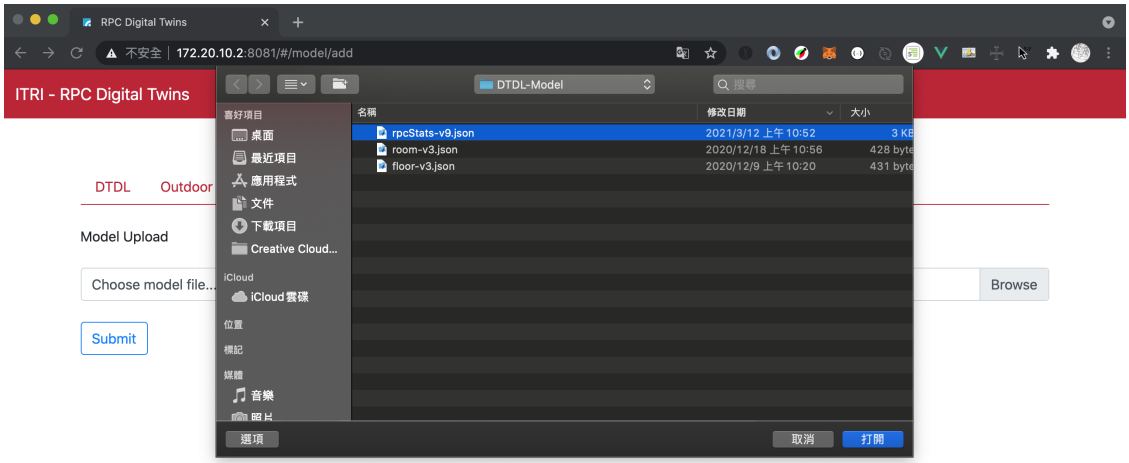
- Select `./Azure-Digital-Twins-for-RPC/DTDL-Model/room-v3.json`



- Click `Submit`
- Upload finish

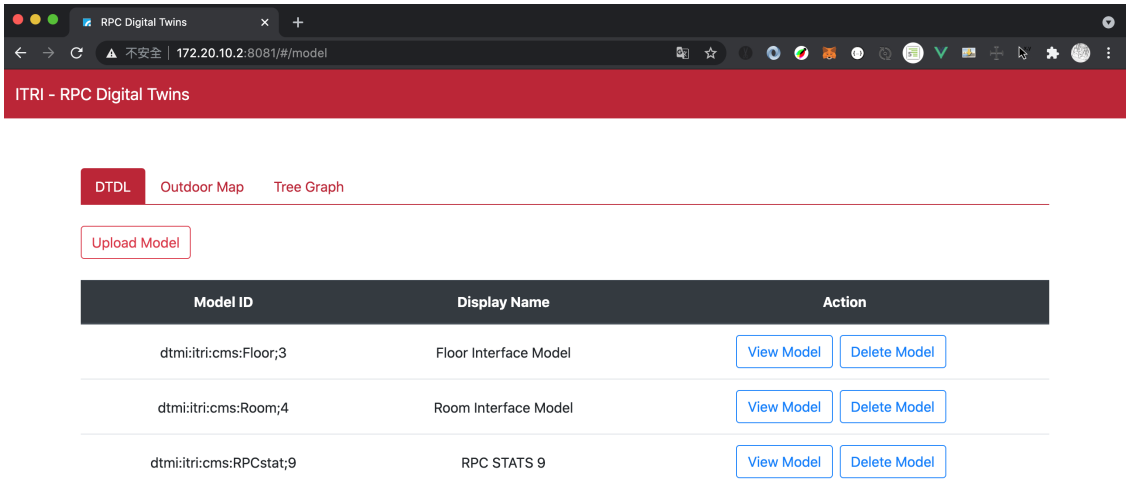


- rpcStats-v9.json
  - Click `Upload Model`
  - Click `Browse`
  - Select `./Azure-Digital-Twins-for-RPC/DTDL-Model/rpcStats-v9.json`



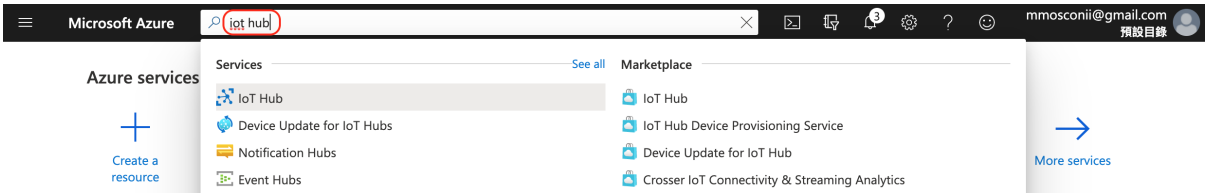
  - Click `Submit`

- Upload finish

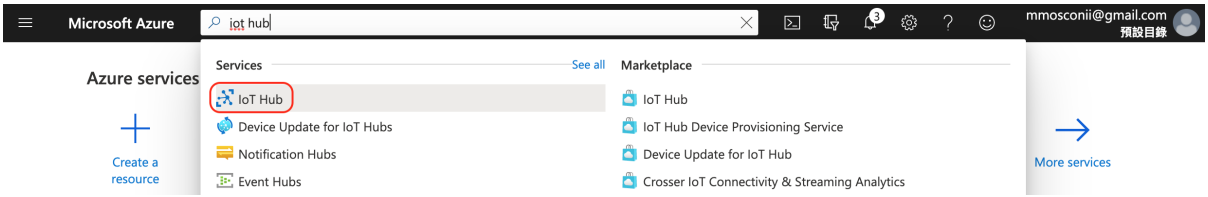


13. Start Device by running rpc_pnp.py

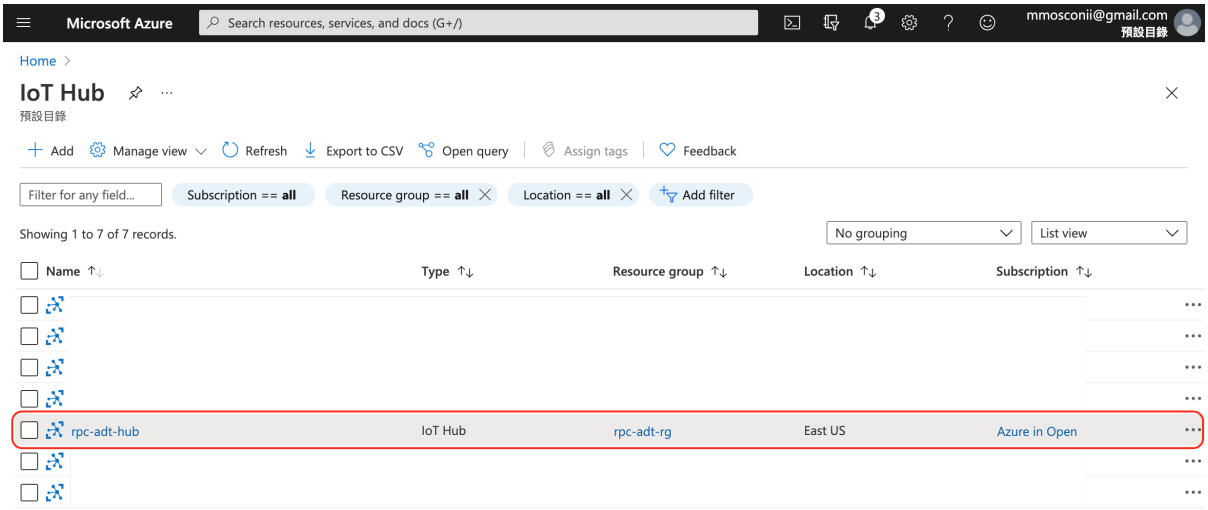
- 13.1 Add `IoT Devices`
  - Search `iot hub`



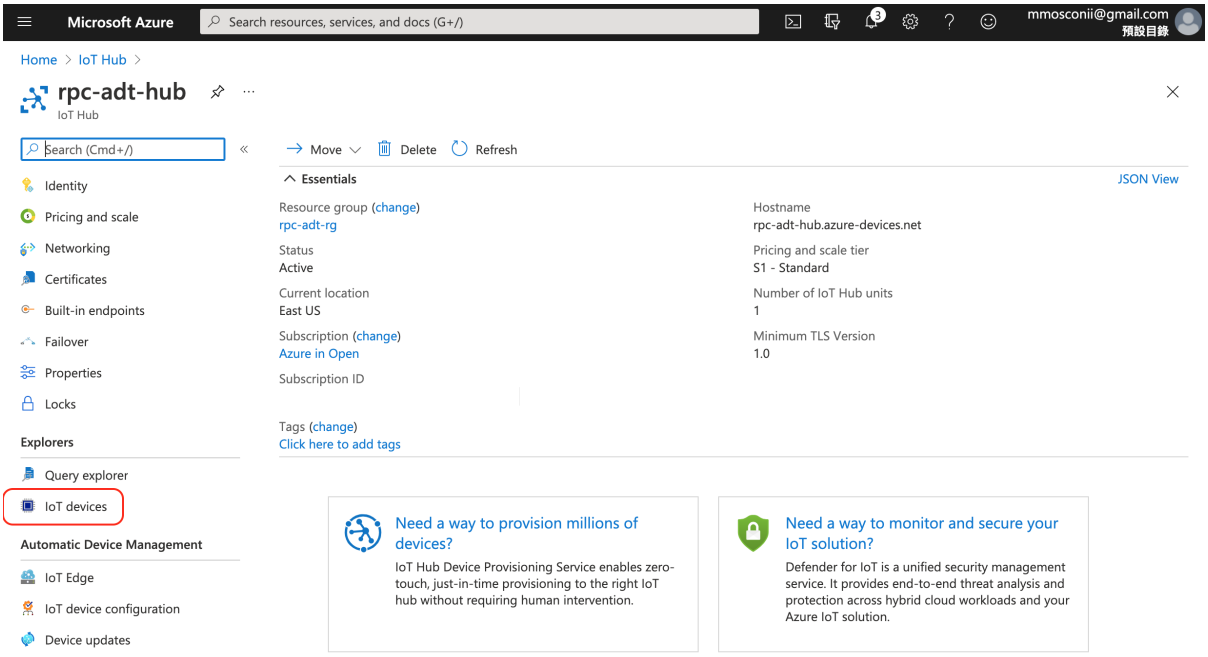
  - Select `IoT Hub`



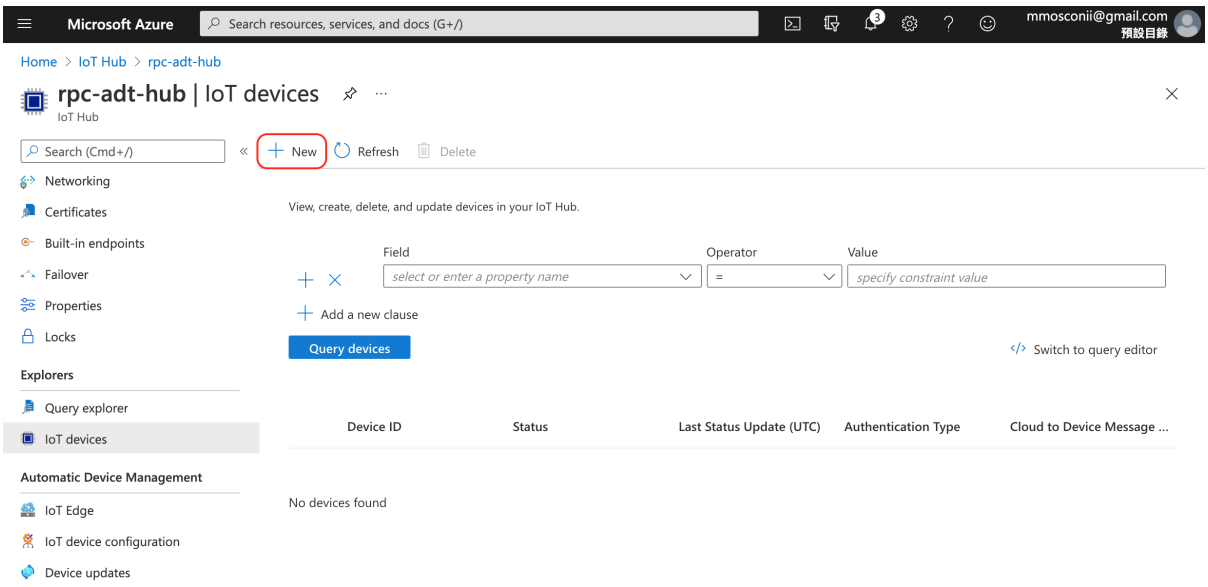
  - Click on the `IoT Hub` created in `Step 8.4`



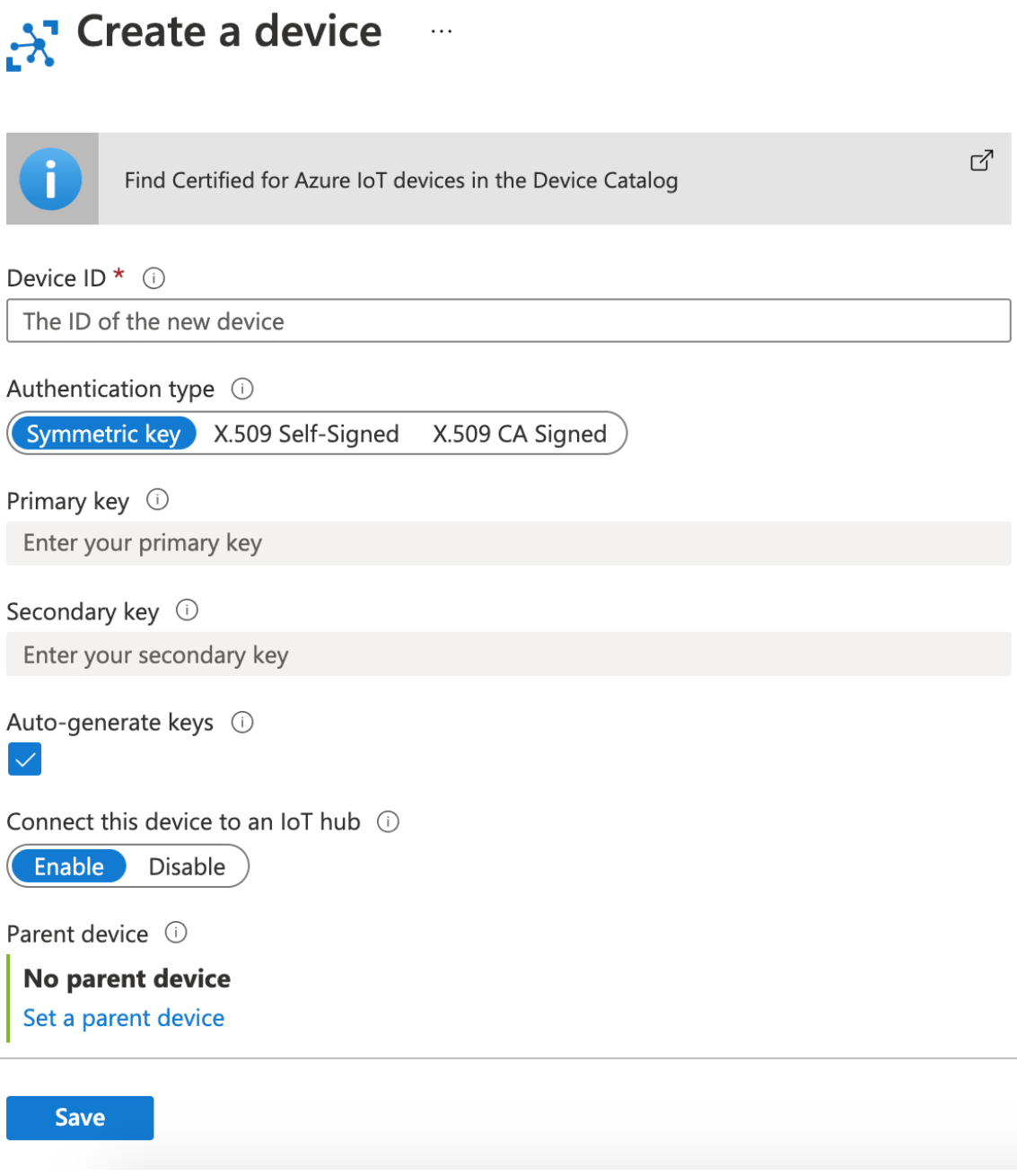
- Click `IoT Devices`



- Click `+ New`

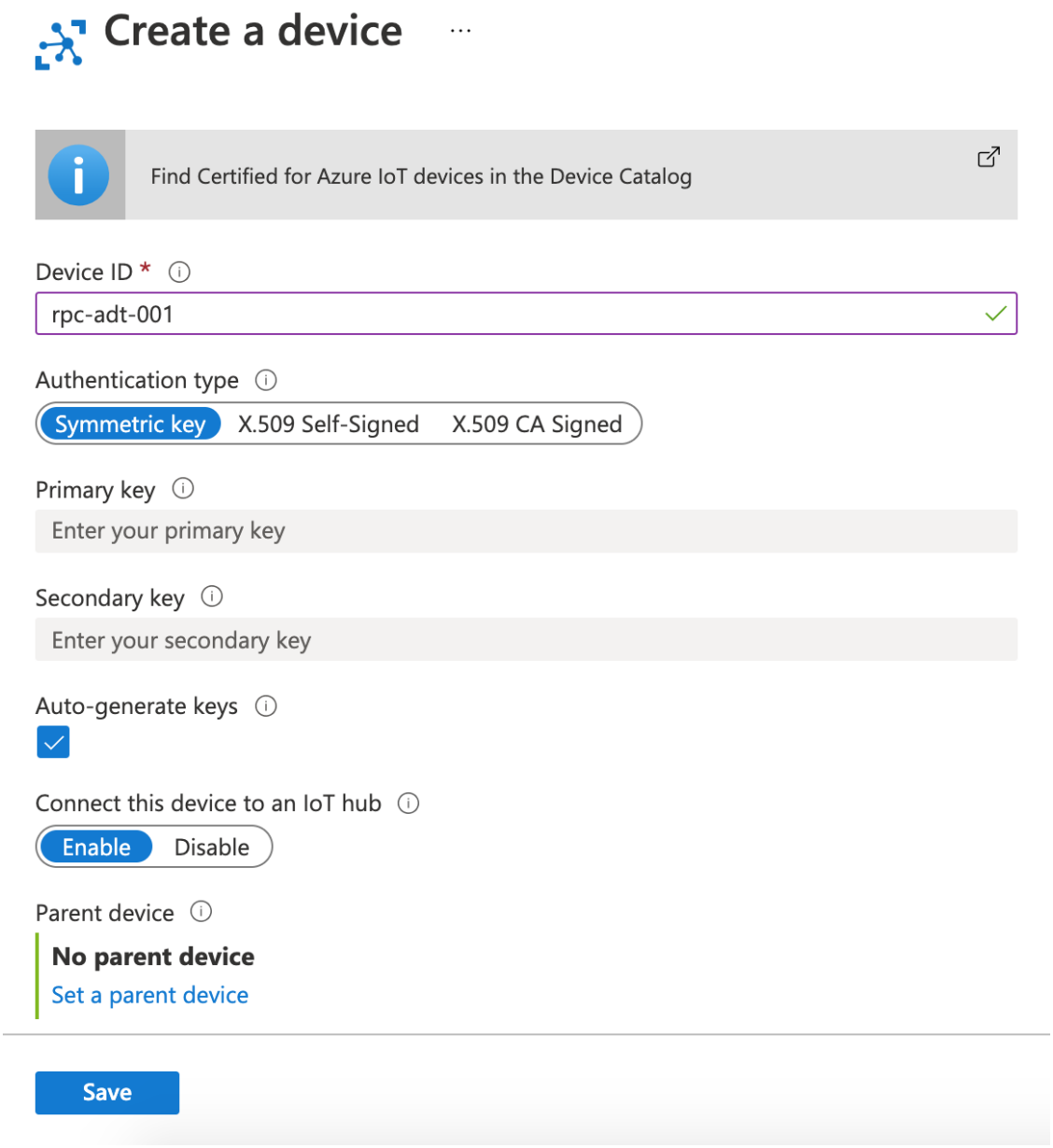


- Input data

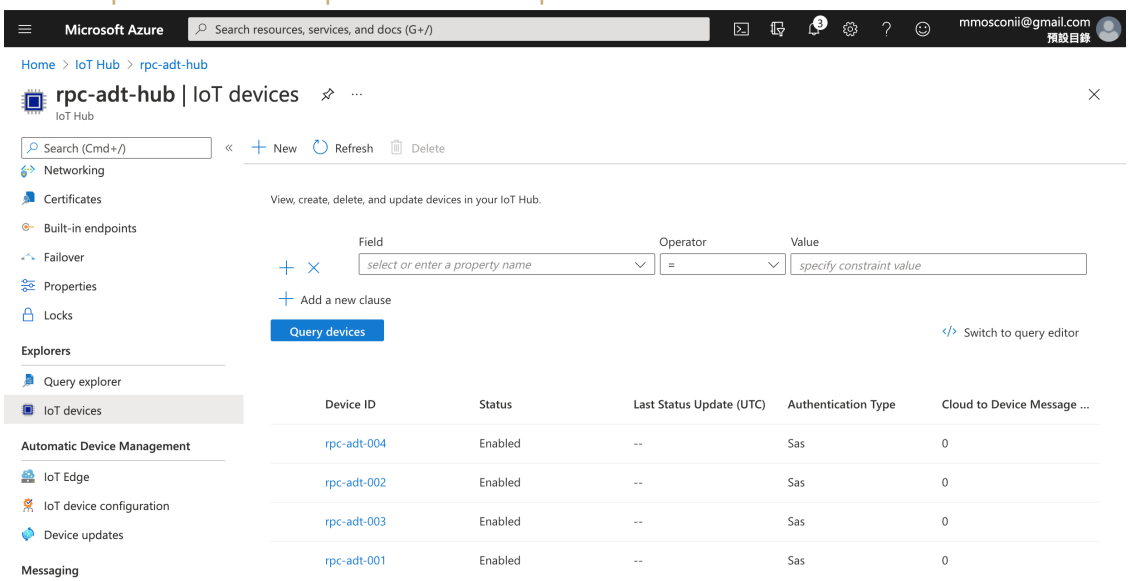


  - `Device ID` field
    - Input `rpc-adt-001`

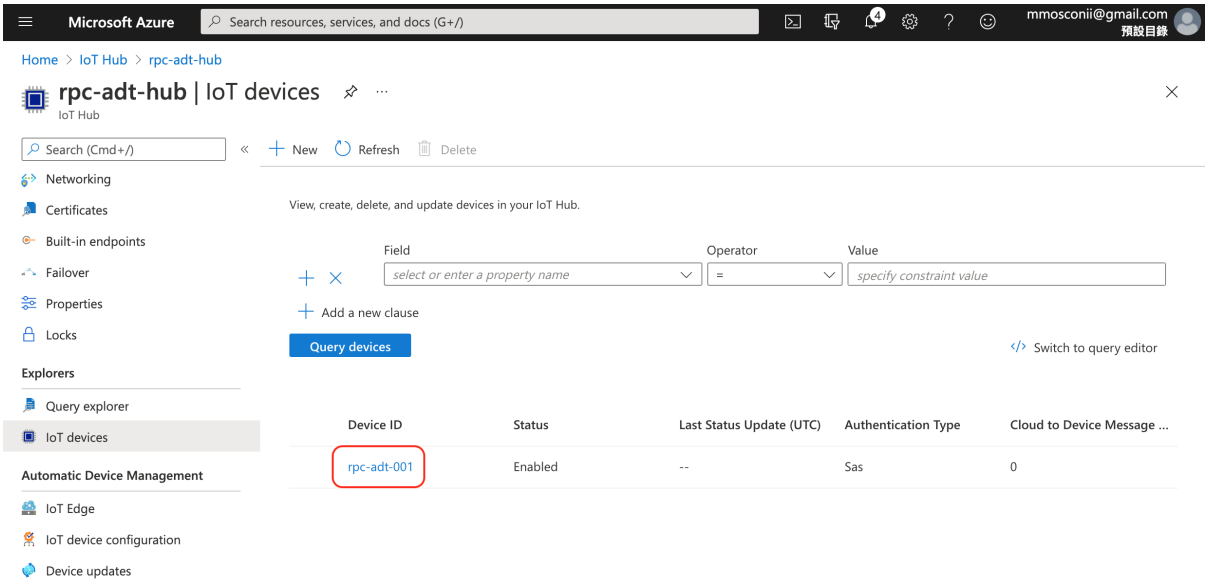
- When the input is complete, please click the Save button



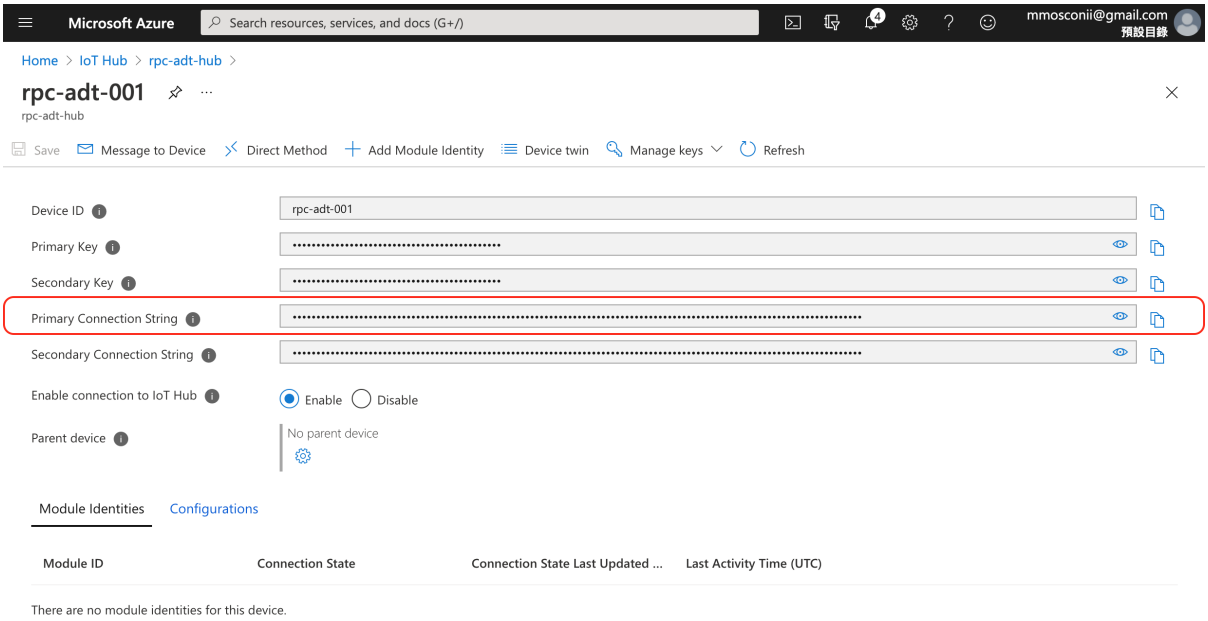
- Create rpc-adt-002、rpc-adt-003、rpc-adt-004



- Click `rpc-adt-001`



- Get `Connection String`



- 13.2 Install Python 3.7.0 on the Plug and Play Device
  - Download And Install
- 13.3 Install Package on the Plug and Play Device
  - Enter the project folder

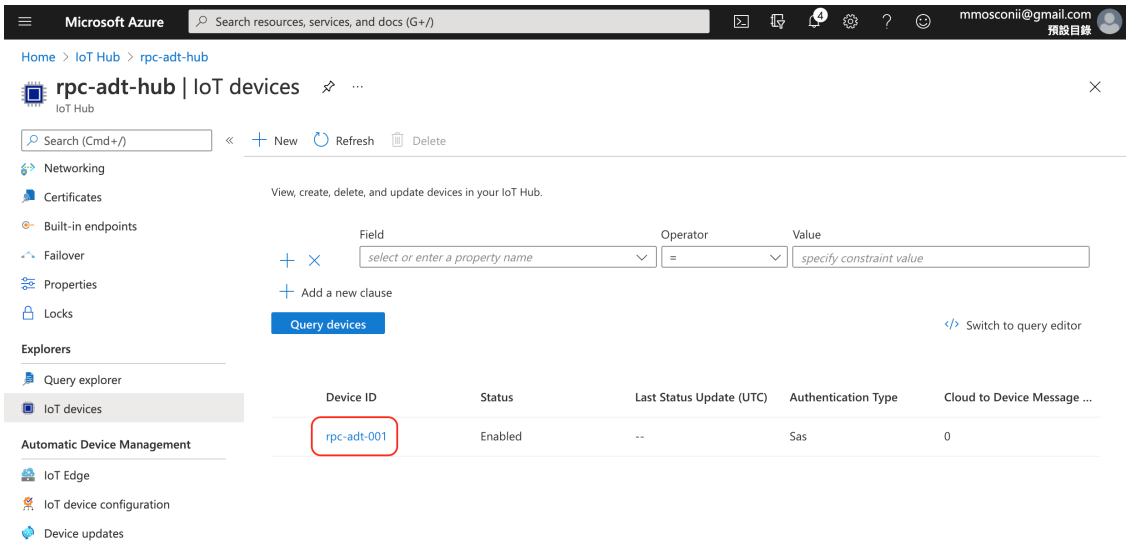
```
$ cd ./Azure-Digital-Twins-for-RPC/PnP-Device/rpc
```

  - Install

```
pip install -r requirements.txt
```

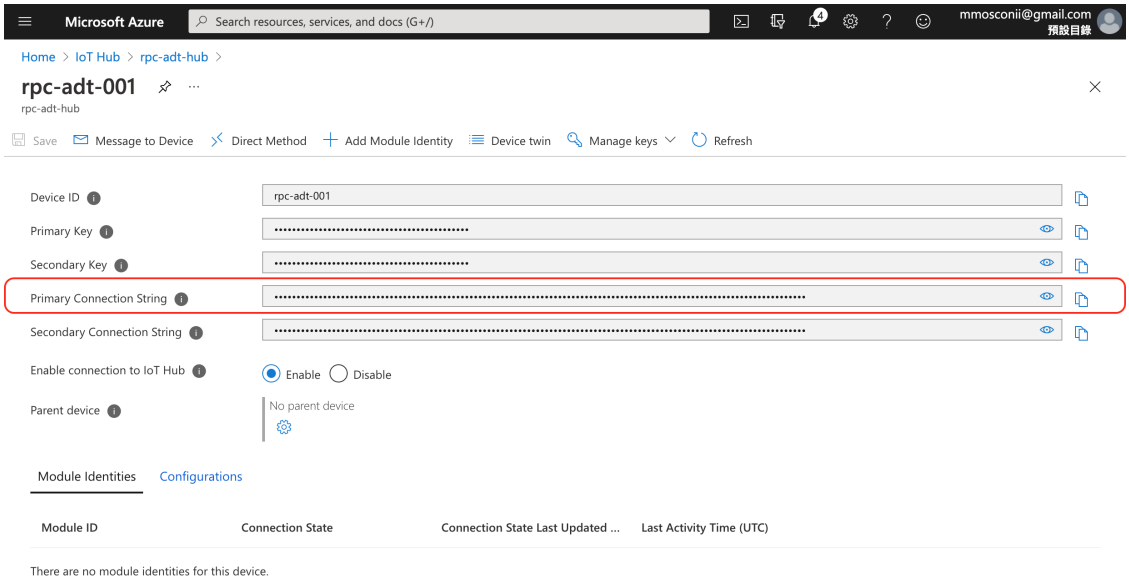
- 13.4 Run code on the Plug and Play Device
  - Change values in rpc_pnp.py
    - Device ID

```
rpc_component_name_01 = 'rpc-adt-001'
```



- IoT Hub connection string

```
IOTHUB_DEVICE_CONNECTION_STRING_DEV=";SharedAccessKeyName=iothubowner"
```

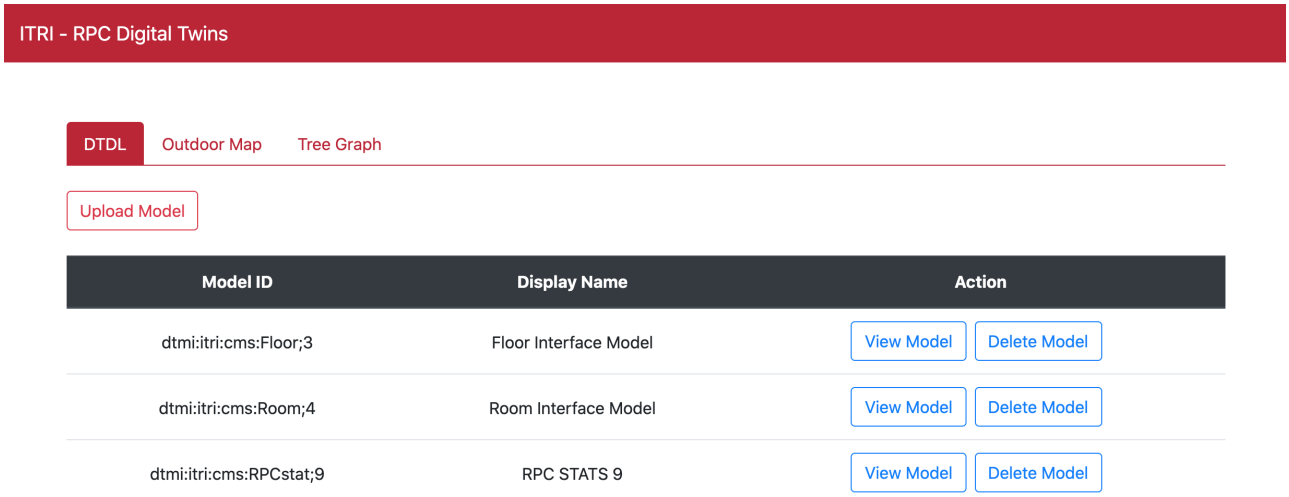


- Run Python

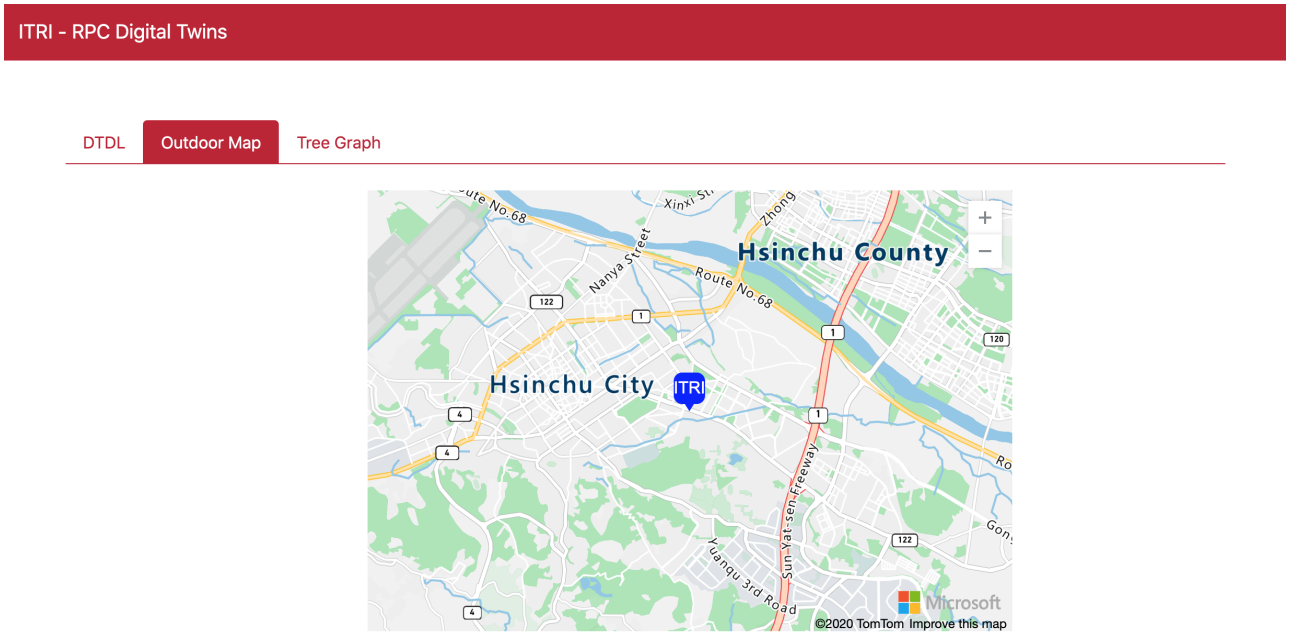
```
$ python3 rpc_pnp.py
```

14. Create Twin

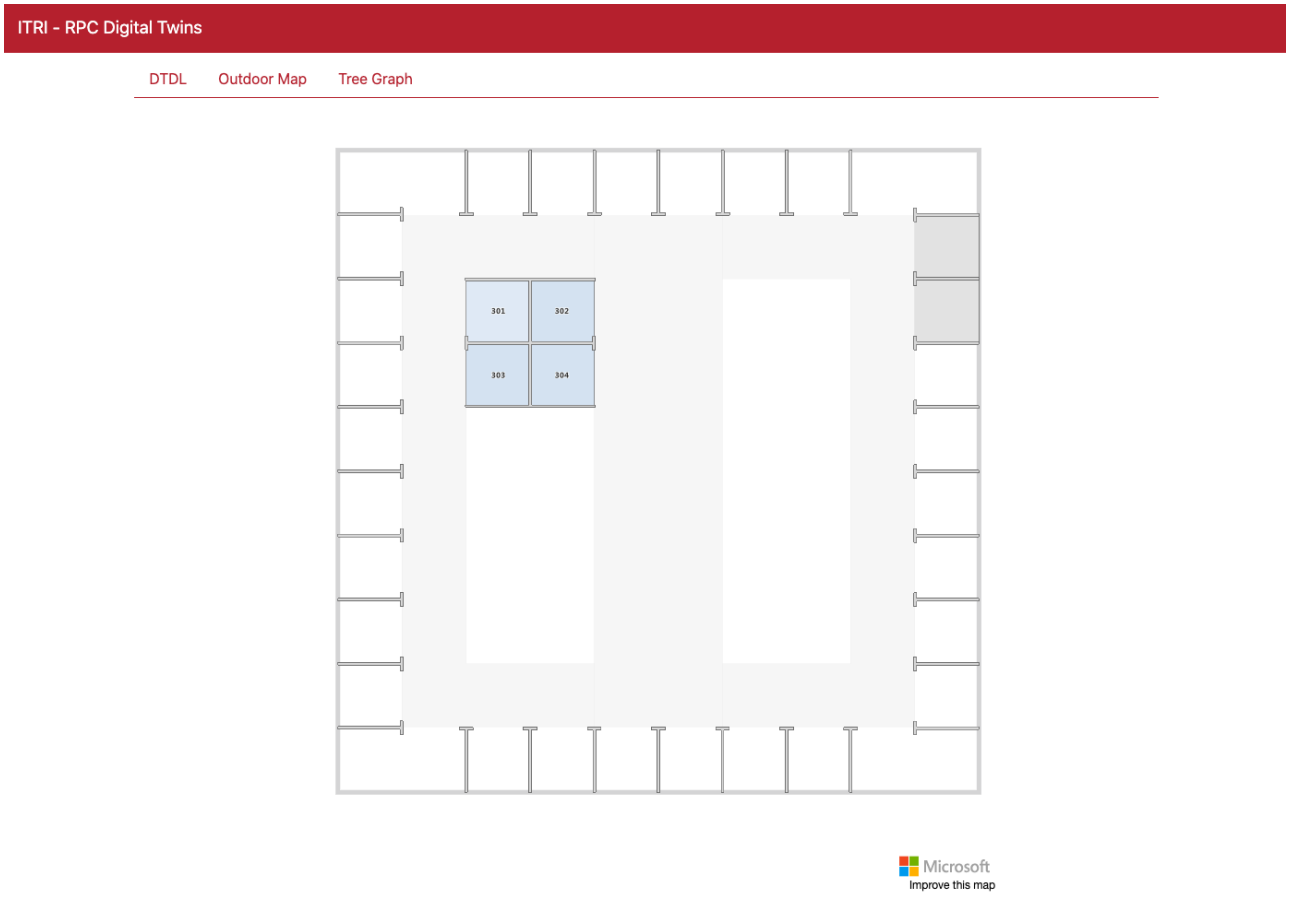
- 14.1 Click `Outdoor Map`



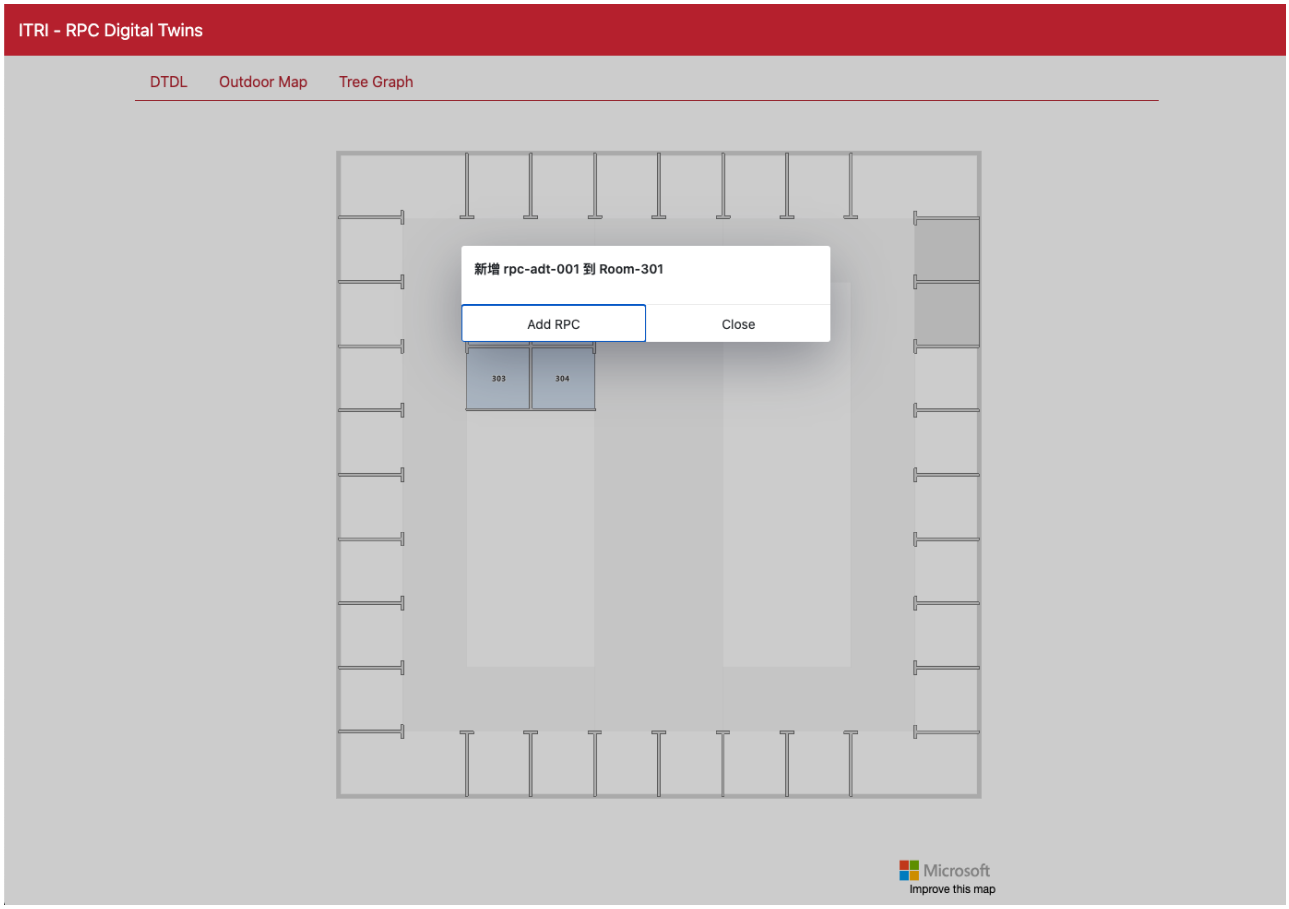
- 14.2 Click `ITRI`



- 14.3 Click 301



- 14.4 Click Add RPC

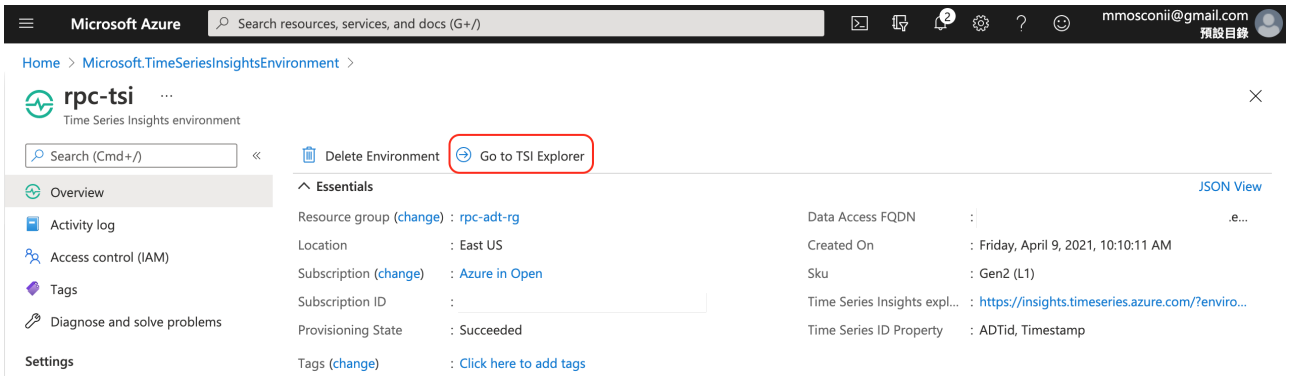


- 14.5 PnP Terminal

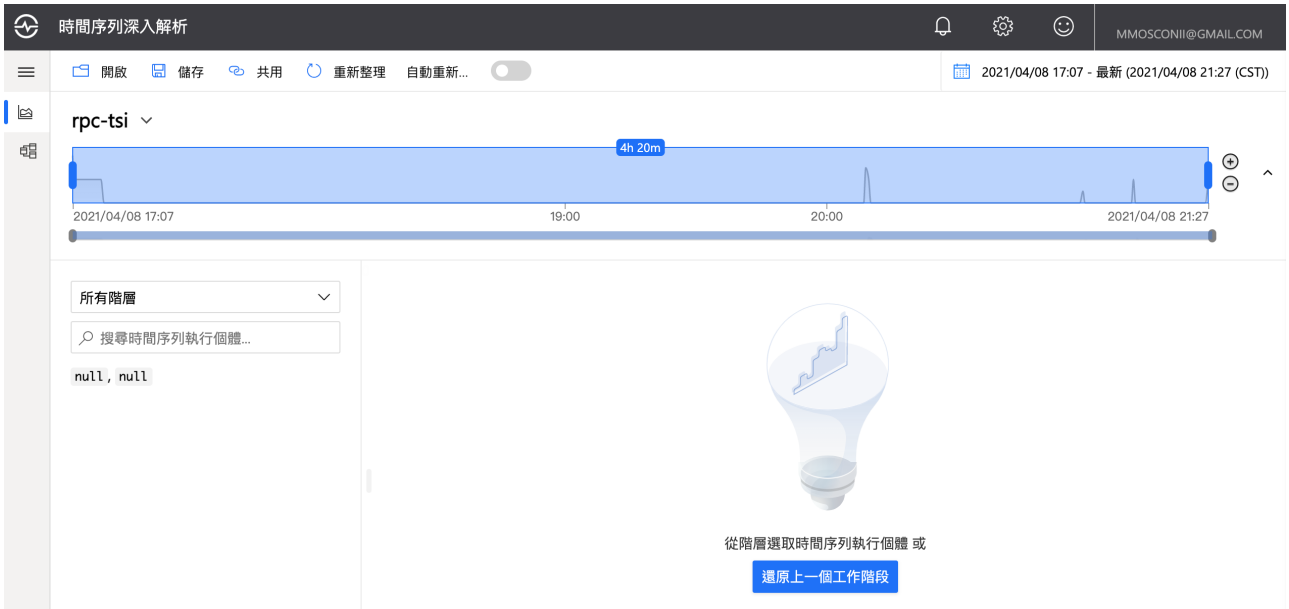
```
rpc — python3 rpc_pnp.py — 126×22

Archer-2:rpc archer$ code .
Archer-2:rpc archer$ python3 rpc_pnp.py
{'$dtId': 'rpc-adt-001', '$metadata': {'$model': 'dtmi:itri:cms:RPCstat;9'}}
Model Id: dtmi:itri:cms:RPCstat;9
Listening for command requests and property updates
Press Q to quit
Sending telemetry from various components

Method callback called with:
methodName = commandtoadd
payload = 19
set fun commandtoadd
now local time now 1618226989
Sent configuration message rpc-adt-001 {"threshole5um": 50000, "threshole3um": 50000, "threshole1um": 50000, "threshole05um":
50000, "timetowatch": 10, "timetosleep": 5, "featureId": 19, "timestamp": 1618226989}
now local time now 1618226999
Sent message rpc-adt-001 {"particle5um": 22158, "particle3um": 38740, "particle1um": 34523, "particle05um": 35857, "rpcAlarm":
 0, "featureId": 19, "timestamp": 1618226999}
```

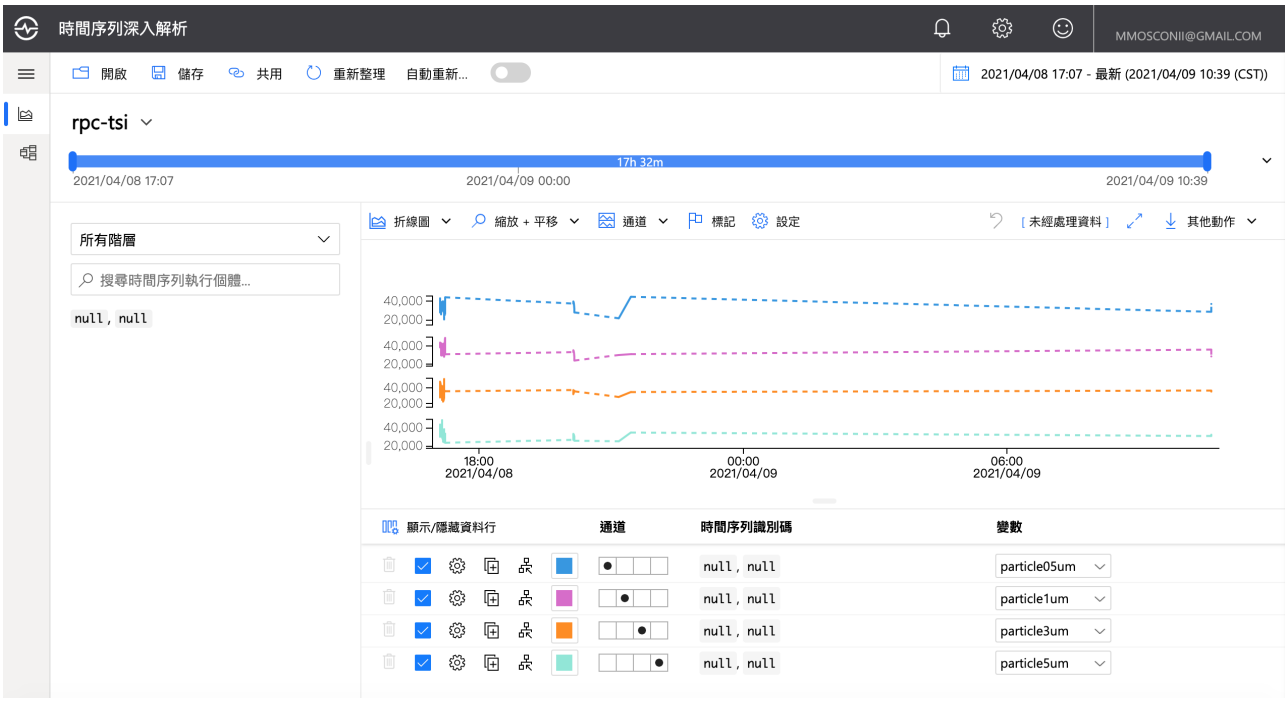
15. Visualize your data in Time Series Insights

- 15.1 Get to TSI Explorer



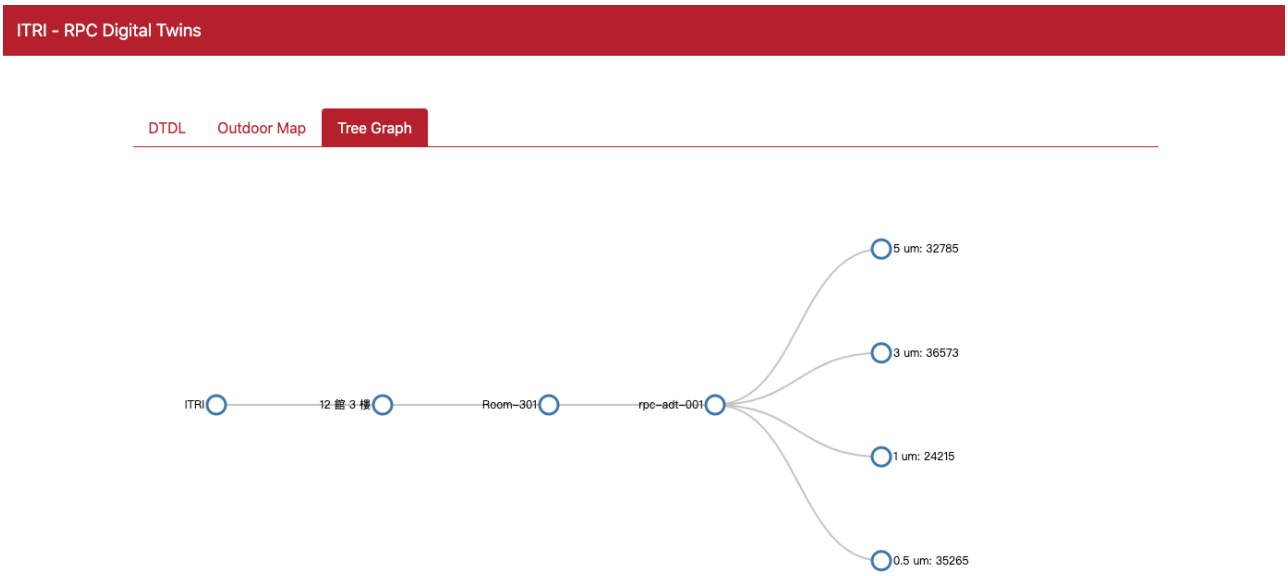
- 15.2 TSI Explorer View





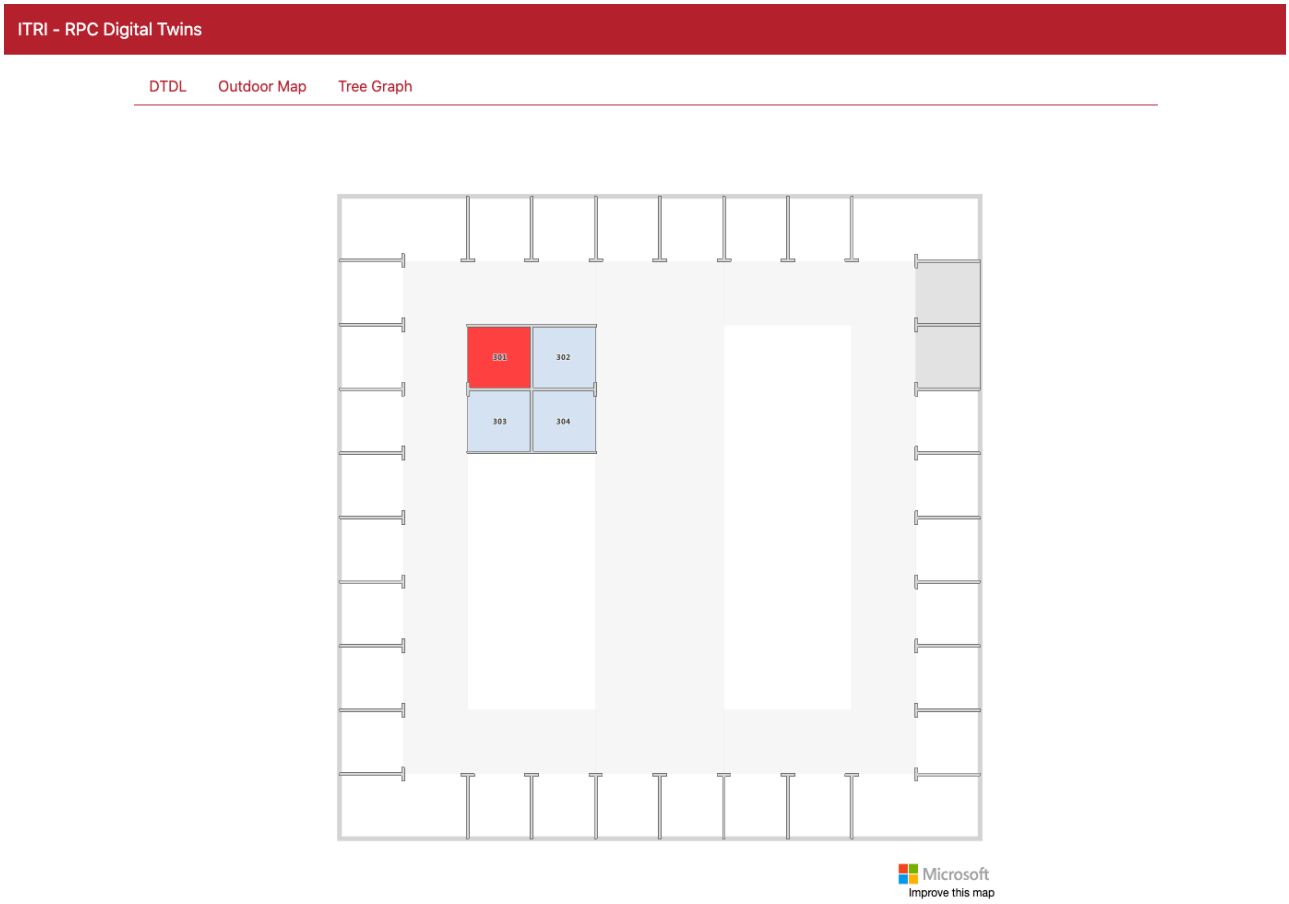
16. View live updates on your dashboard

- Click `Tree Graph`



17. View live updates on your map

- Alarm


ITRI - RPC Digital Twins
DTDL
Outdoor Map
Tree Graph
301
302
303
304
Microsoft
Improve this map