Decouple Power Apps with Azure Integration

- Praveen

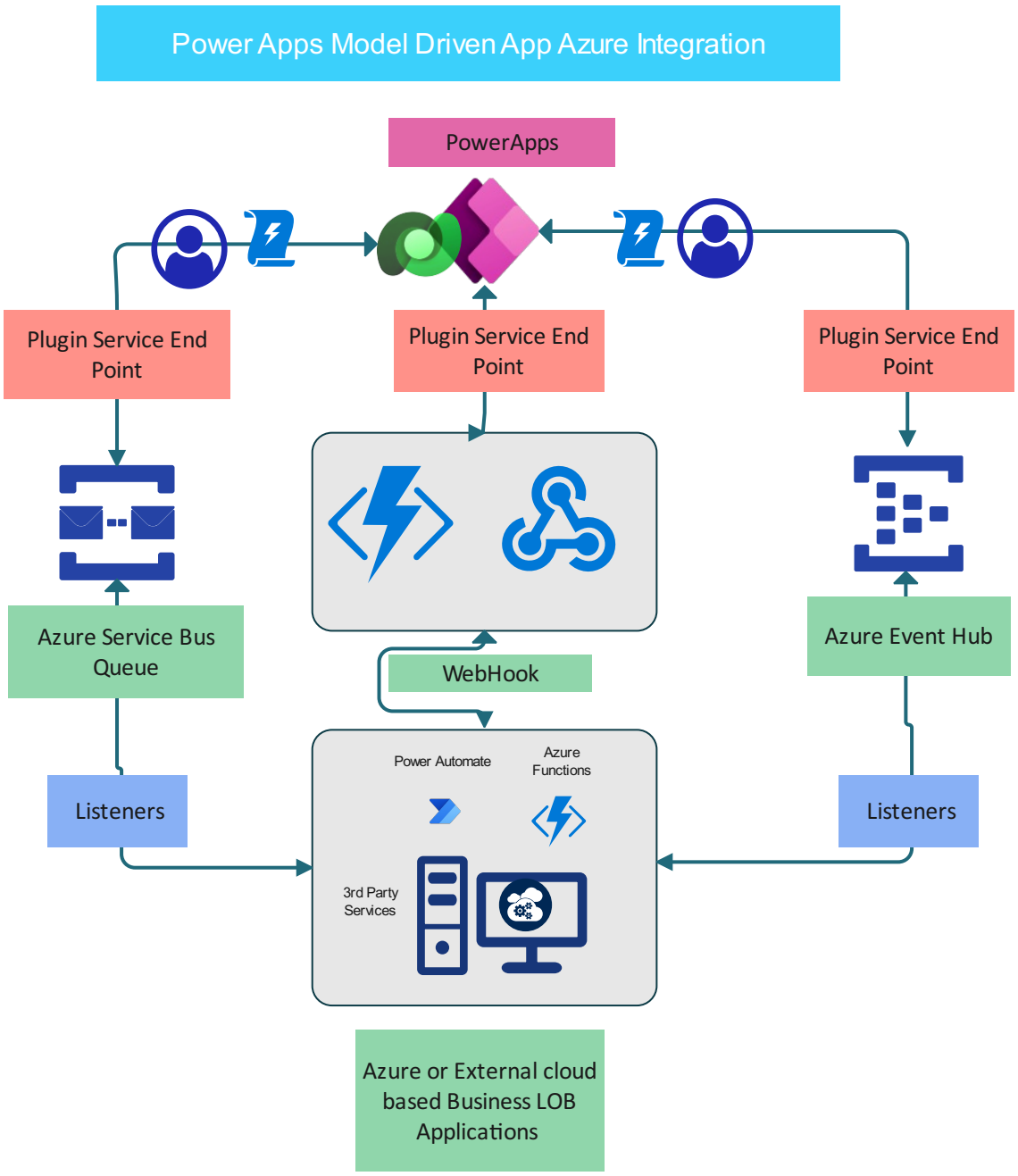
Contents

Decouple Power Apps with Azure Integration

- Praveen

By integrating with Azure Service Bus / Azure Event Hub / Azure Functions,

- **We are able to decouple power apps and improve the scalability and dependability of the application and services.**
- **Line-of-business (LOB) apps from outside the Dataverse can safely and effectively communicate with one other.**
- **For maintaining corporate data synchronization between numerous Dataverse systems or other Dataverse servers.**



High Level Steps of integration

Create Azure Service Bus / Azure Event Hub/ Webhook

You can create an **event hub or service bus** in Azure either through API programming or interactively by using the Azure portal. Either way, after creating your event hub you must obtain a copy of the connection string and provide that string when registering the Azure service endpoint detailed in the next section.

With Microsoft Dataverse, you can send data about events that occur on the server to a web application using **webhooks**. Webhooks is a lightweight HTTP pattern for connecting Web APIs and services with a publish/ subscribe model. Webhook senders notify receivers about events by making requests to receiver endpoints with some information about the events.

Register a new Service Endpoint / new Webhook

Use the Plug-in Registration tool (PRT) to register the service endpoint. When filling out the PRT registration form specify a contract type of Event Hub or service bus. For the message body format, you can choose XML or JSON. You must provide the connection string obtained when you created the event hub or service bus.

In the Plug-in Registration tool there is a new Register New WebHook option to select. When you register a WebHook you must provide three items of information - Name, Endpoint URL & Authentication (HttpHeader, WebhookKey- used for azure functions, HttpQueryString).

Register a Step & Start Listening

Dataverse needs to know the exact operation (table and message combination) that, when processed, would cause the Azure-aware plug-in to execute. Since you are creating an event hub/Service bus/Webhook, this operation would be related to the processing of Azure service bus/eventhub/ functions data in particular. You must register a step for the Azure-aware plug-in in the execution pipeline.

Trigger the action

Perform an operation in Dataverse that would cause the plug-in to execute. This is the same operation (table and message combination) that you registered the plug-in step for in the previous section of this topic. You can perform the intended operation by using the power apps model driven app or web application or through application code accessing the Azure web services.

Verification

You can check the related system job in the Dataverse environment and look for a status of Succeeded. If you find a status of Failed, use the status information to identify the possible cause of the failure. You can then recheck the configurations of both systems

External cloud LOB Application Consumer listening the Actions for performing business cases

You can check the related system job in the Dataverse environment and look for a status of Succeeded. If you find a status of Failed, use the status information to identify the possible cause of the failure. You can then recheck the configurations of both systems

Azure EventHub

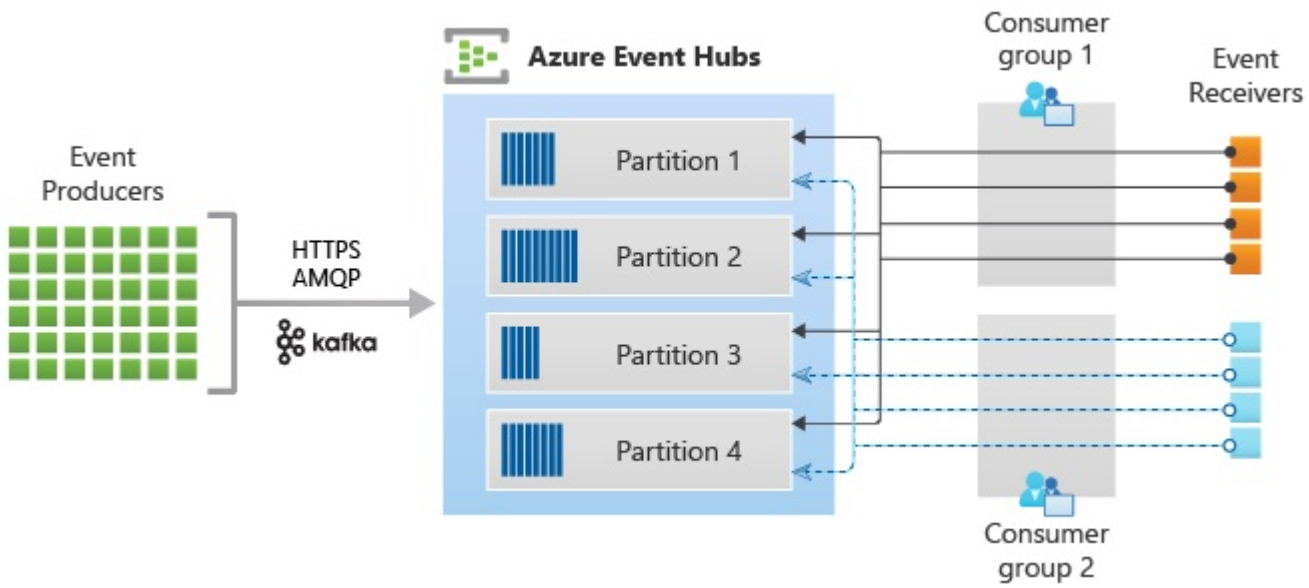
Key features

Paas

Batch Processing

Real time Processing

Azure Data Lake Storage Gen2



Integrations

Azure SQL

Machine Learning

Stream Analytics

Power BI

Azure storage

Scenarios

Application Logging

Anomaly detection

Analytics Pipeline

Transaction Processing

Archiving Data

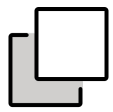
Azure ServiceBus

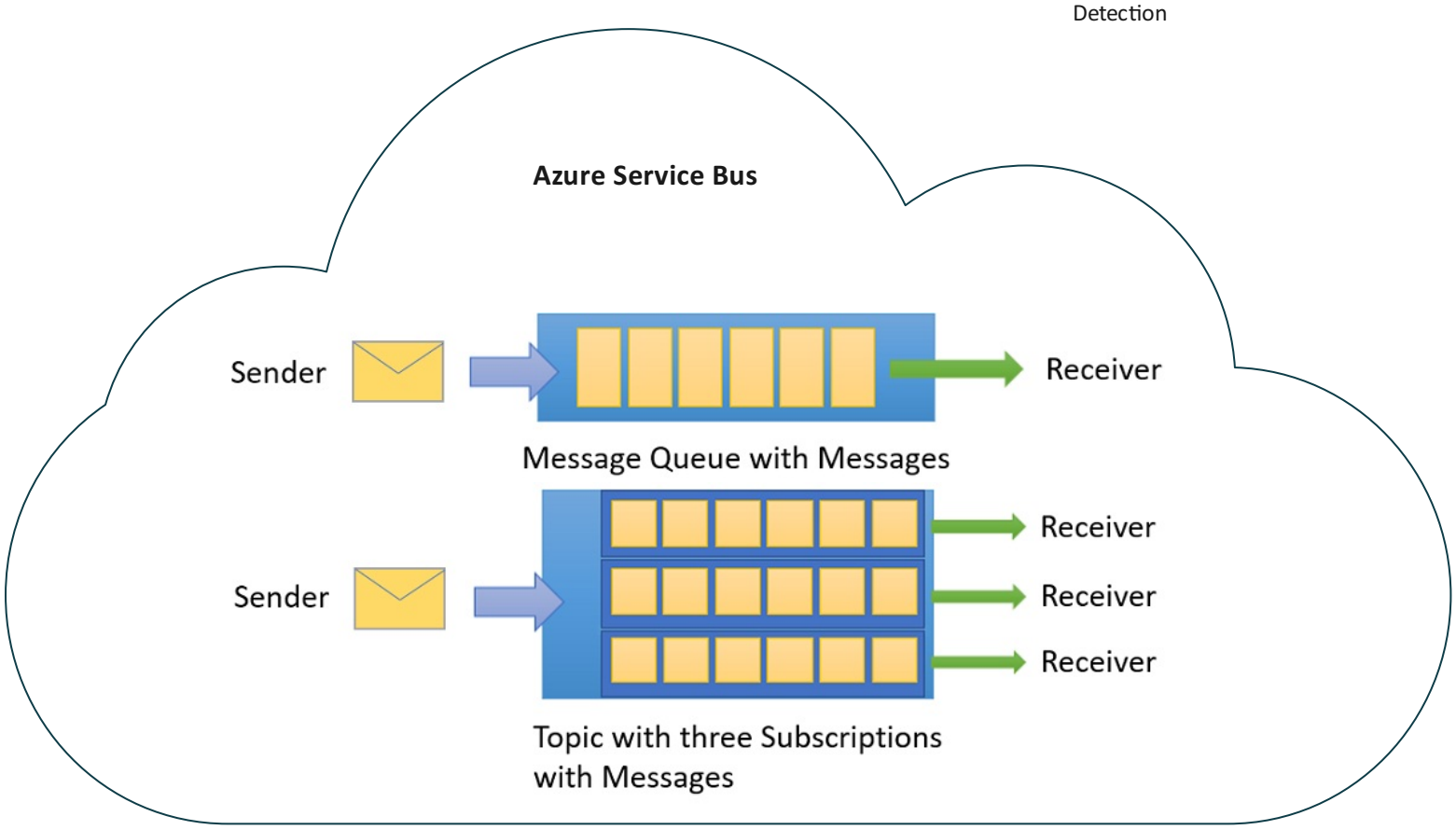
Key features


Topics


Queue

AMQP 1.0


Duplicate Detection



Integrations


Azure Event Grid


Logic Apps


Azure function App


Power Platform


Stream Analytics

Scenarios


Messaging


Decoupling Applications


Load Balancing

Transaction Processing

When deciding between the WebHook model and the Azure Service Bus integration, here are some items to keep in mind:

- **Azure Service Bus works for high scale processing, and provides a full queueing mechanism if Dataverse is pushing many events.**
- **Webhooks can only scale to the point at which your hosted web service can handle the messages.**
- **Webhooks enables synchronous and asynchronous steps. Azure Service Busonly allows for asynchronous steps.**
- **Webhooks send POST requests with JSON payload and can be consumed by any programming language or web application hosted anywhere.**
- **Both Webhooks and Azure Service Bus can be invoked from a plug-in or custom workflow activity.**

	Azure Service Bus	Event Hubs
Definition	Azure Service Bus is a fully managed enterprise message broker with message queues and publish-subscribe topics	Azure Event Hubs is a big data streaming platform and event ingestion service. It can receive and process millions of events per second. Data sent to an event hub can be transformed and stored by using any real-time analytics provider or batching/ storage adapters.
Scenario and usages	Messaging Decouple applications Load balancing Topics and subscriptions Transactions Message sessions	Anomaly detection (fraud/outliers) Application logging Analytics pipelines, such as clickstreams Live dashboards Archiving data Transaction processing User telemetry processing Device telemetry streaming
Key features	Queues Topics Messaging Sessions Auto-Forwarding Dead-Lettering Scheduled Delivery Message Deferral Transactions Filtering & Actions Auto Delete on Idle Duplicate Deletion Supports standard Advanced Message Queuing Protocol (AMQP) 1.0 and HTTP/ REST protocols.	Fully managed PaaS Support for real-time and batch processing Capture event data - Long term retention (azure blob storage or azure data lake storage) Scalable
Integration	Event Grid Logic Apps Azure Functions Power Platform Dynamics 365 Azure Stream Analytics	Azure SQL DB Azure Storage Stream analytics Power BI Machine learning
Pricing	Basic Standard Premium more info: https://azure.microsoft.com/ en-us/pricing/details/service-bus/	Basic Standard Premium more info: https://azure.microsoft.com/en-us/pricing/details/ event-hubs/

Installation of Plugin Registration Tool & Power Platform Tools in Visual Studio

We can either use PRT or Power Platform Tools in Visual studio to create service endpoints. New Service End Point creation and New step registration will have have the same interface in both PRT and visual studio power platform tools. We will be using PRT tool in the following demos.

Install Plugin Registration Tool

Steps to install PRT using Power Platform CLI

Microsoft Power Platform CLI is a simple, one-stop developer CLI that empowers developers and ISVs to perform various operations in Microsoft Power Platform related to environment lifecycle, authentication, and work with Microsoft Dataverse environments, solution packages, portals, code components, and more.

We will install Plugin Registration tool using Power Platform CLI.

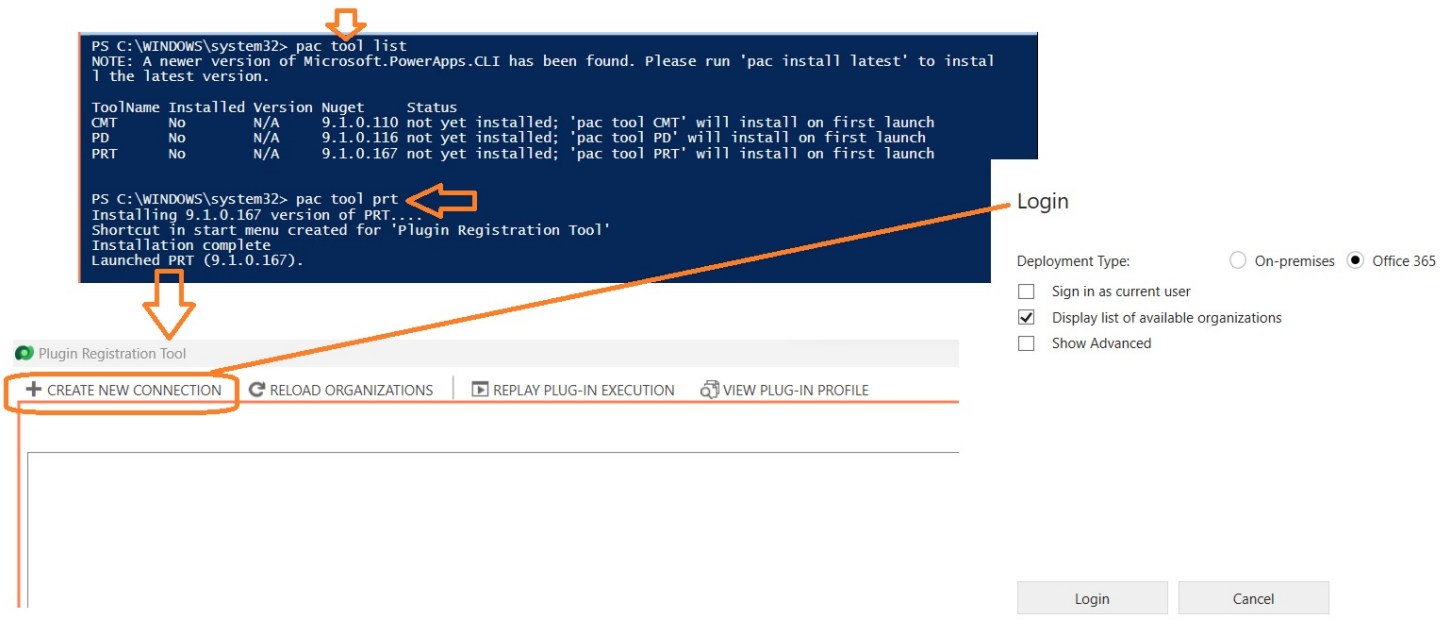
If power platform cli is not installed in your system , install Using the below Link
https://learn.microsoft.com/en-us/power-platform/developer/cli/introduction#install-using-power-platform-tools-for-visual-studio-code
https://learn.microsoft.com/en-us/power-platform/developer/cli/introduction#install-power-platform-cli-for-windows

Once CLI is installed, open the CLI and type the below command to check avaliable/installed tools for the system,
pac tool list

If the PRT tools is not installed on the system, type the below command to install the tool in the system,
pac tool prt

pac tool prt will install the Plugin registration tool in the sytem. Once installed you can open the PRT tool and create new connection.

Click on Office 365 and select Display list of avaliable organizations and click on Login
Once Logged in Select the environment where you want to create service endpoint and proceed to next section

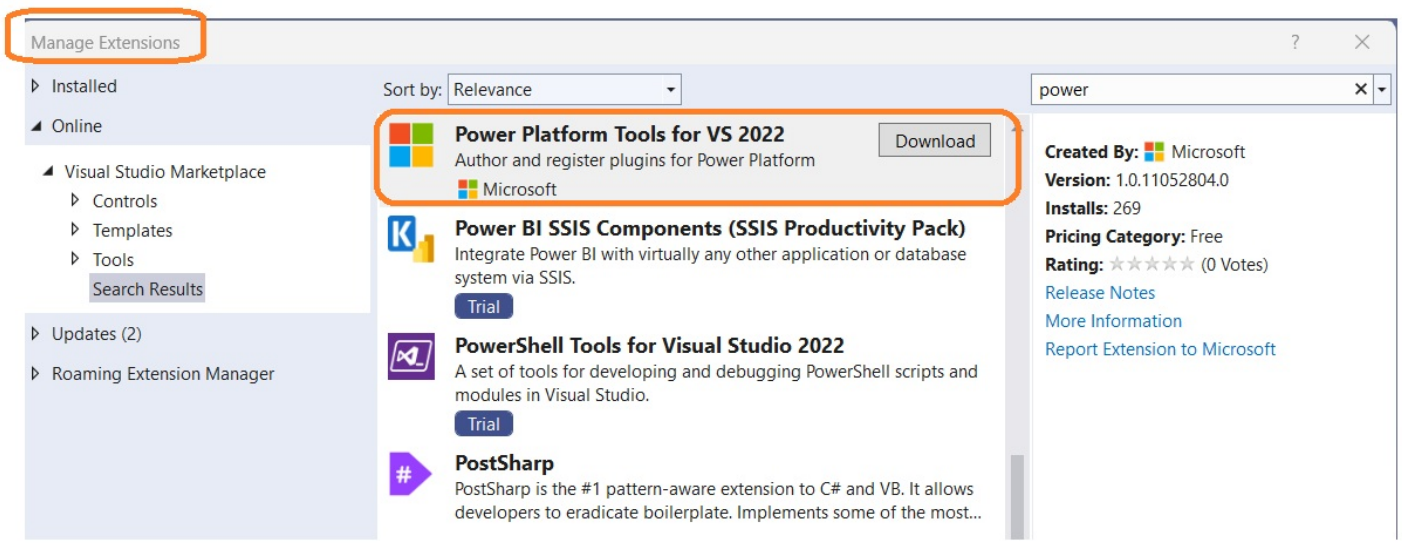


Install Power Platform Tools in Visual Studio

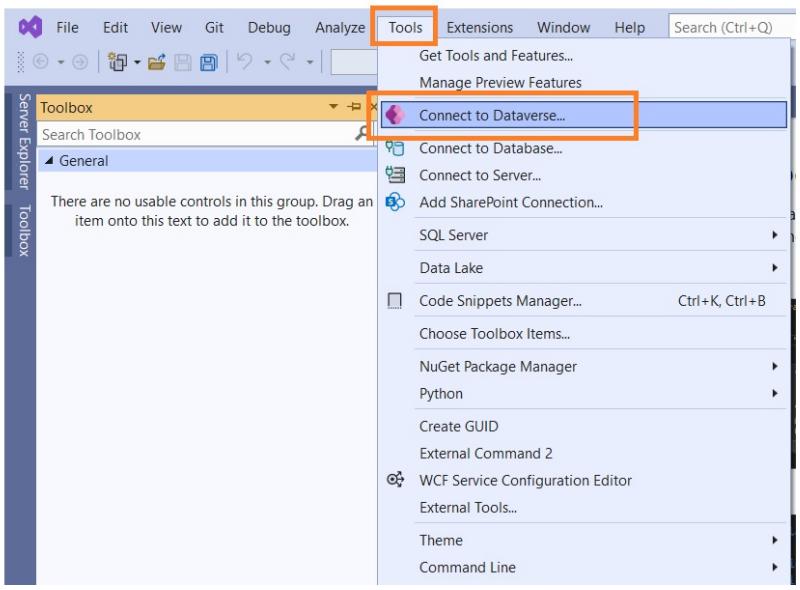
Steps to install PowerPlatformTools in VisualStudio

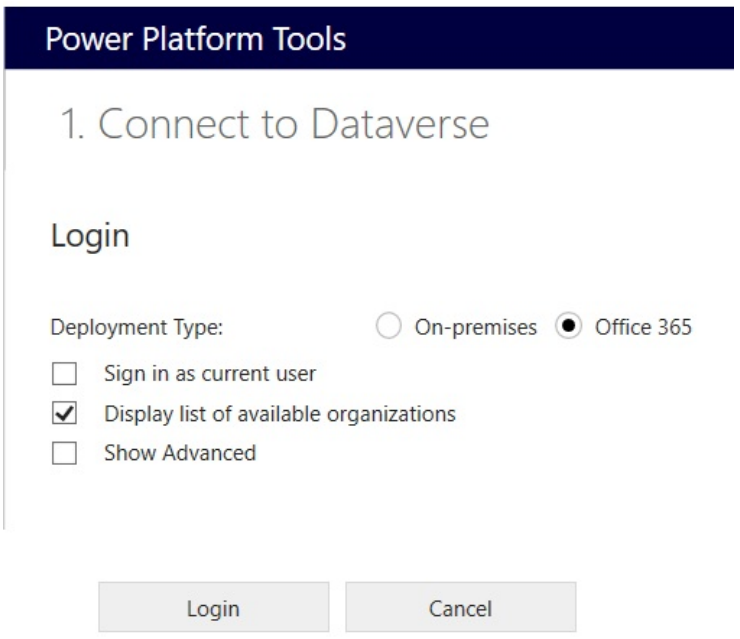
This is a optional part to cover the installtion steps for Power platform tools in visual studio. We can either user PRT or Power Platform Tools for visual studio.

Open Visual Studio 2019 or 2022
CLick on Extensions --> Select Manage Extensions
Under Manage Extension --> search Power Platform Tools in the visual studio marketplace and click on download



Once the tool gets downloaded, CLick Tools
Under Tools --> select COnnect to Dataverse --> this will open the Power Platform Tools
Click on Office 365 and select Display list of avaliable organizations and click on Login
Once Logged in Select the environment where you want to create service endpoint and proceed to next section





Intergrate with Azure Service Bus

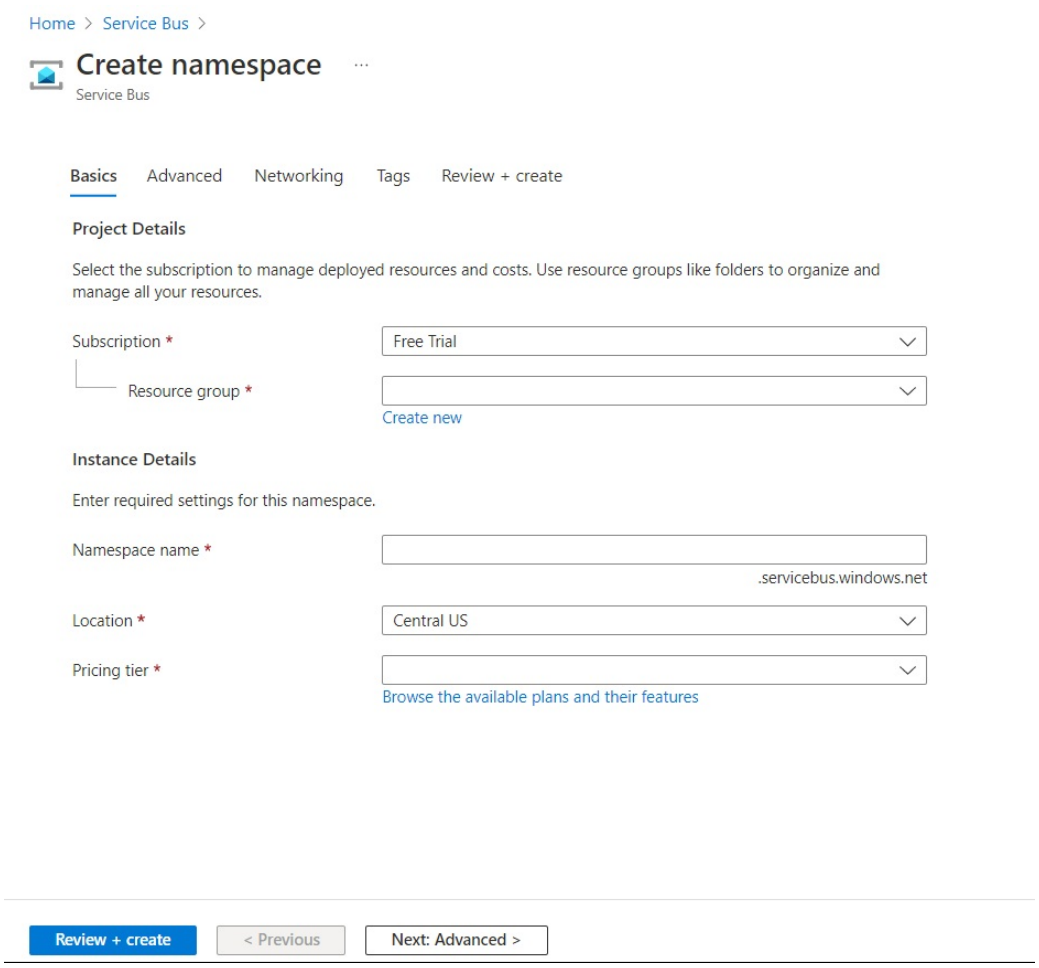
Sections:

- Creation of Azure Service bus in Azure Portal
- Installation of Plugin Registration Tool via Power Platform CLI
- New Registration of Service Endpoint for Service Bus
- New Step Registration aganist Service Bus Service Endpoint
- Trigger the action using the model-driven app
- Verify the action message in Azure Service bus Queue Explorer
- Listen to the Azure Service bus Queue message via Azure Function App
- Listen to the Azure Service bus Queue message via Power Automate

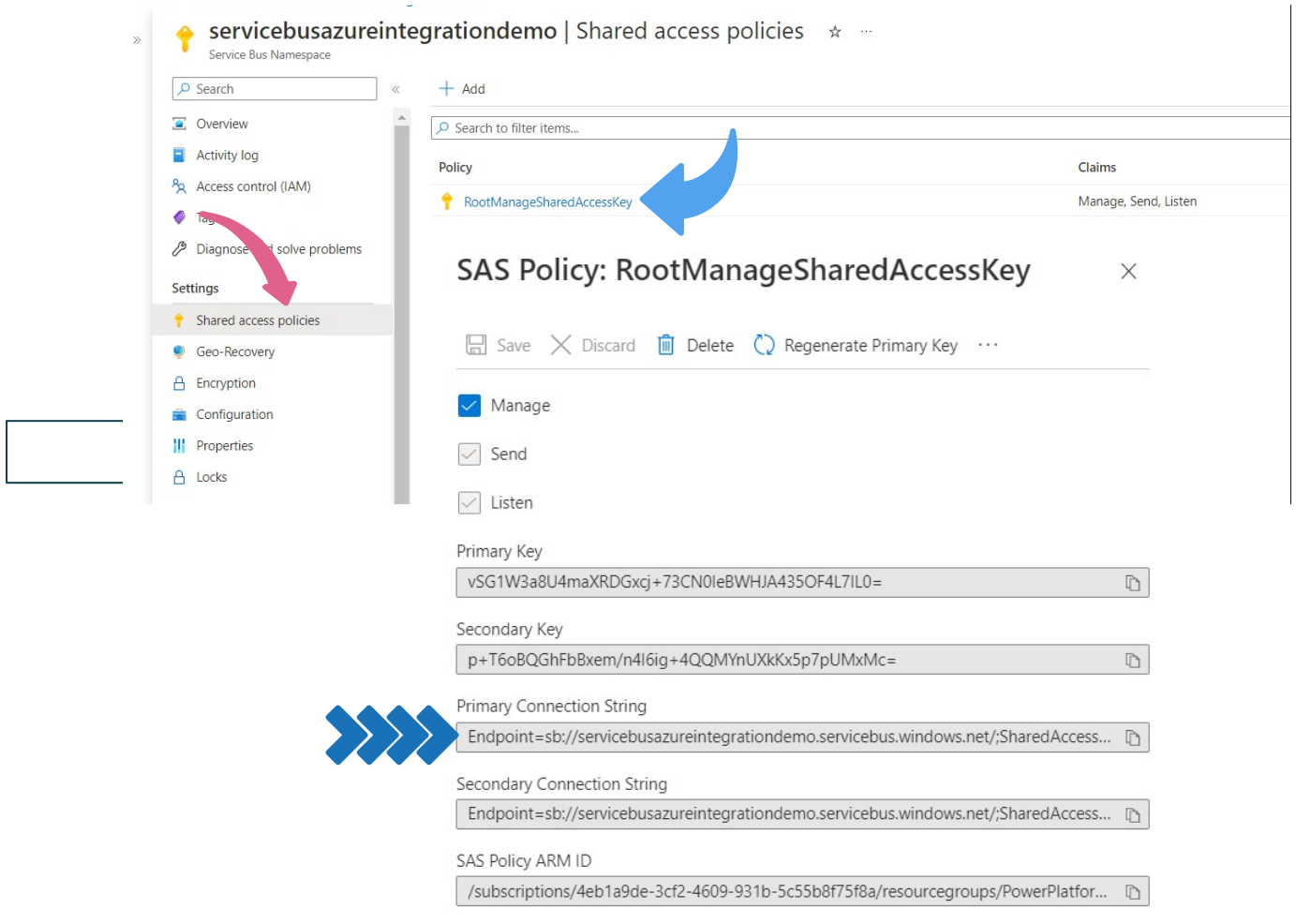
Service Bus Queue Creation in Azure Portal

Steps for Creating Azure Service Bus NameSpace & Queue

Go to Azure portal and search for Azure Service Bus.
Select Service Bus --> Click on Create NameSpace.
Provide the subscription , resource group, Namespace name, Location, Pricing tier details and click on create.

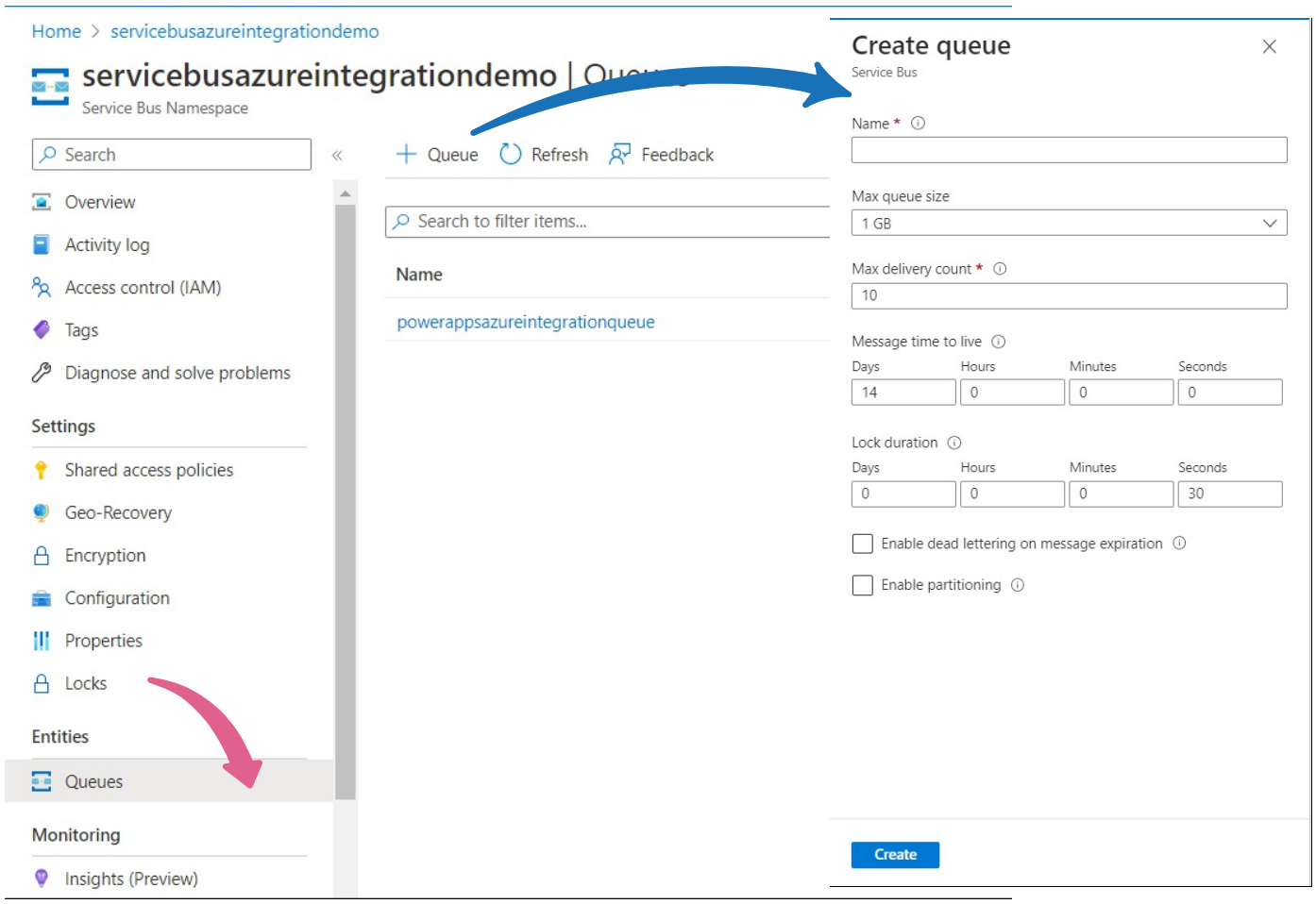


Once Azure Service Bus Name space is created, go to service bus.
Under Settings --> Click on Shared access policies and copy the connection strings and paste it in notepad. These connection string will be used while creating the service end point in Plugin Registration tool.

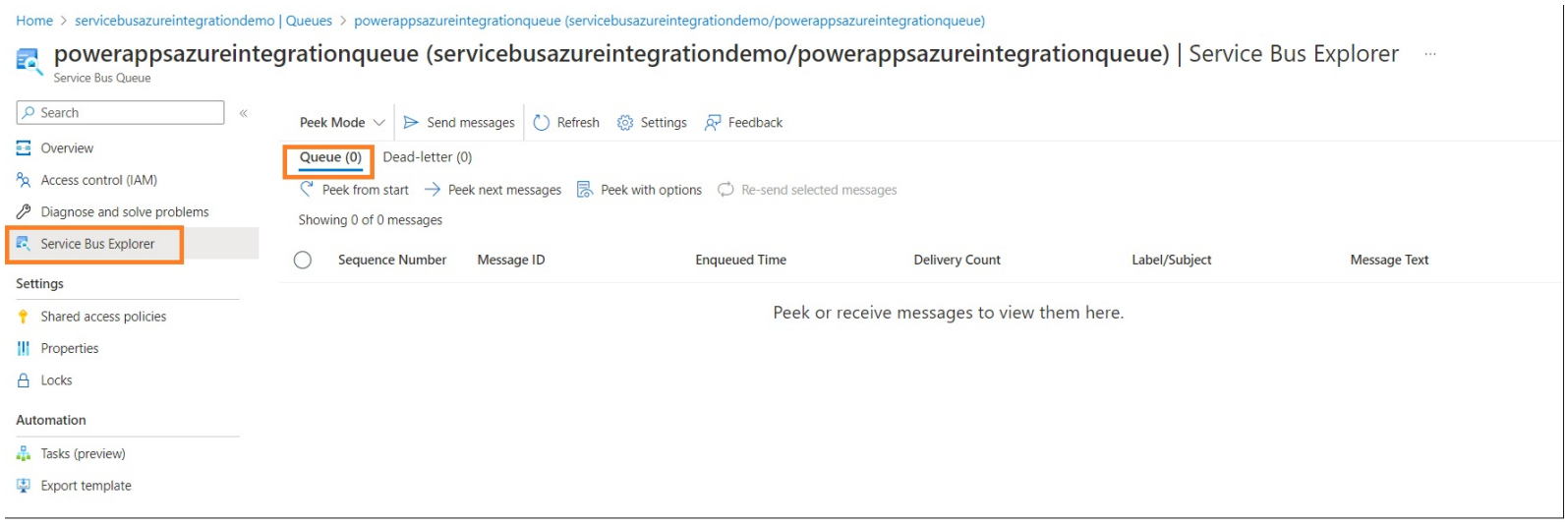


For the demo, we will be creating the Service Bus Queue in this service bus namespace. So whenever some one creates a new item in the account entity this service bus queue will receive the json queue message about the newly created account item details via service endpoint and step.

Under Entities --> Click on Queue
Provide the name, and leave the other settings to default and create the queue.



Once the queue is created, go inside the queue.
We can see Service Bus Explorer is avaliable. with this we can verfiy the new queue message.



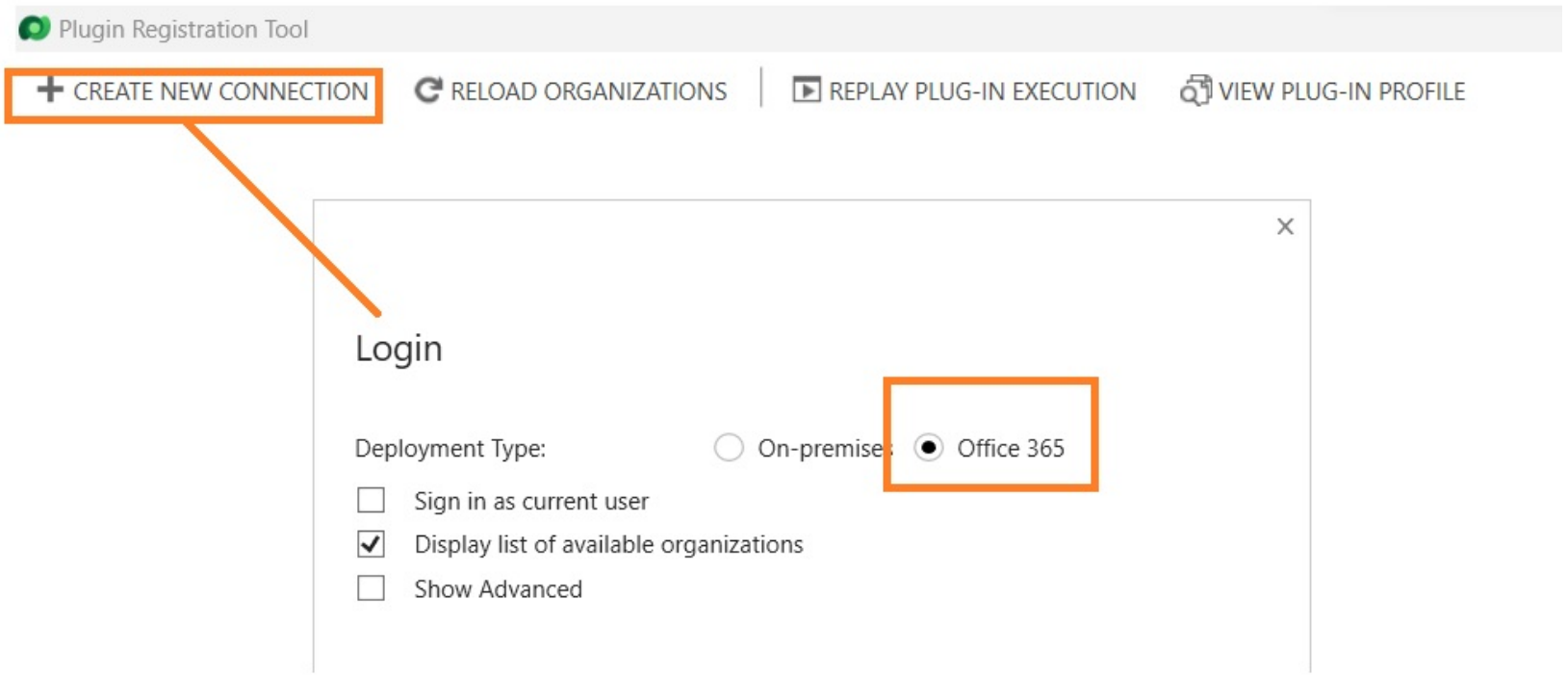
Now the service bus queue creation is completed. we will open the PRT tool and register the service end point and step to integrate the dataverse to Azure Service Bus.

Register Service Endpoint for Azure Service Bus

Steps for creating new service endpoint for azure service bus in PRT

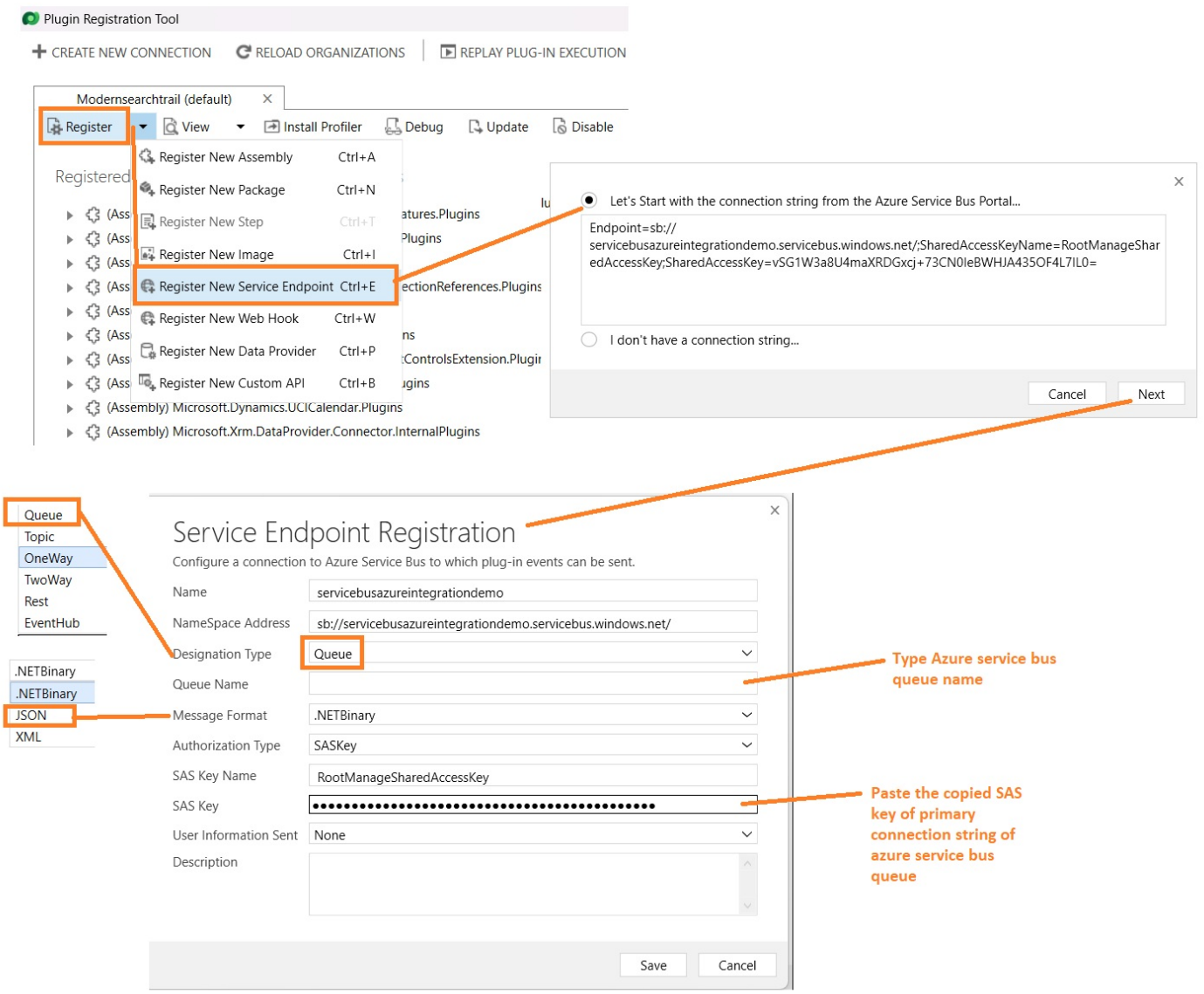
Open PRT Tool and click on Create New Connection
Click on Office 365 and select Display list of avaliable organizations and click on Login
Once Logged in Select the environment where you want to create service endpoint and proceed to next section.

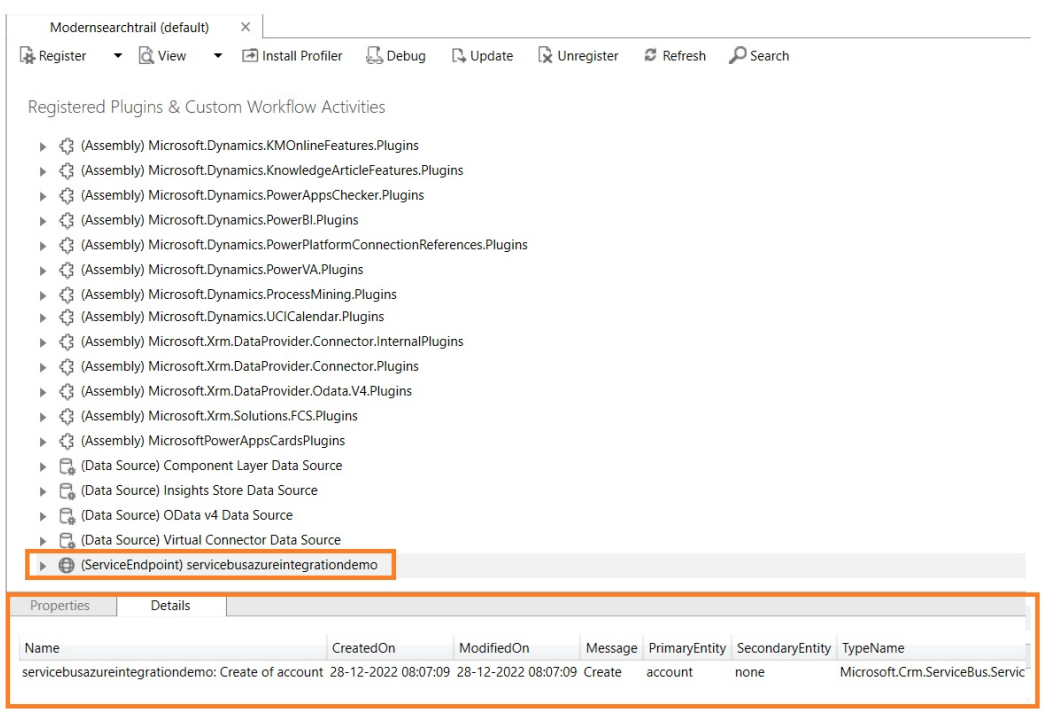
(For this demo we will be creating the service end point connection for the default environment. so we have selected default environment)



Once the New Connection is registered aganist the enviornment.
Select Register --> Click on Register New Service Endpoint
In the Service Endpoint PopUp --> copy the connection string of azure service bus copied previously and click on Next.

This will open up the Service Endpoint Registration.
In the Serice Endpoint Registration popup, PRT will pick most of the details and update it to respective section based on the connection string. we need to select the destination type as "Queue" & Message format as "JSON" & verify the other fields.
Once verified click on Save.
This will create the Azure service bus service endpoint registration for this environment.
Next we need Create new step against this service endpoint.



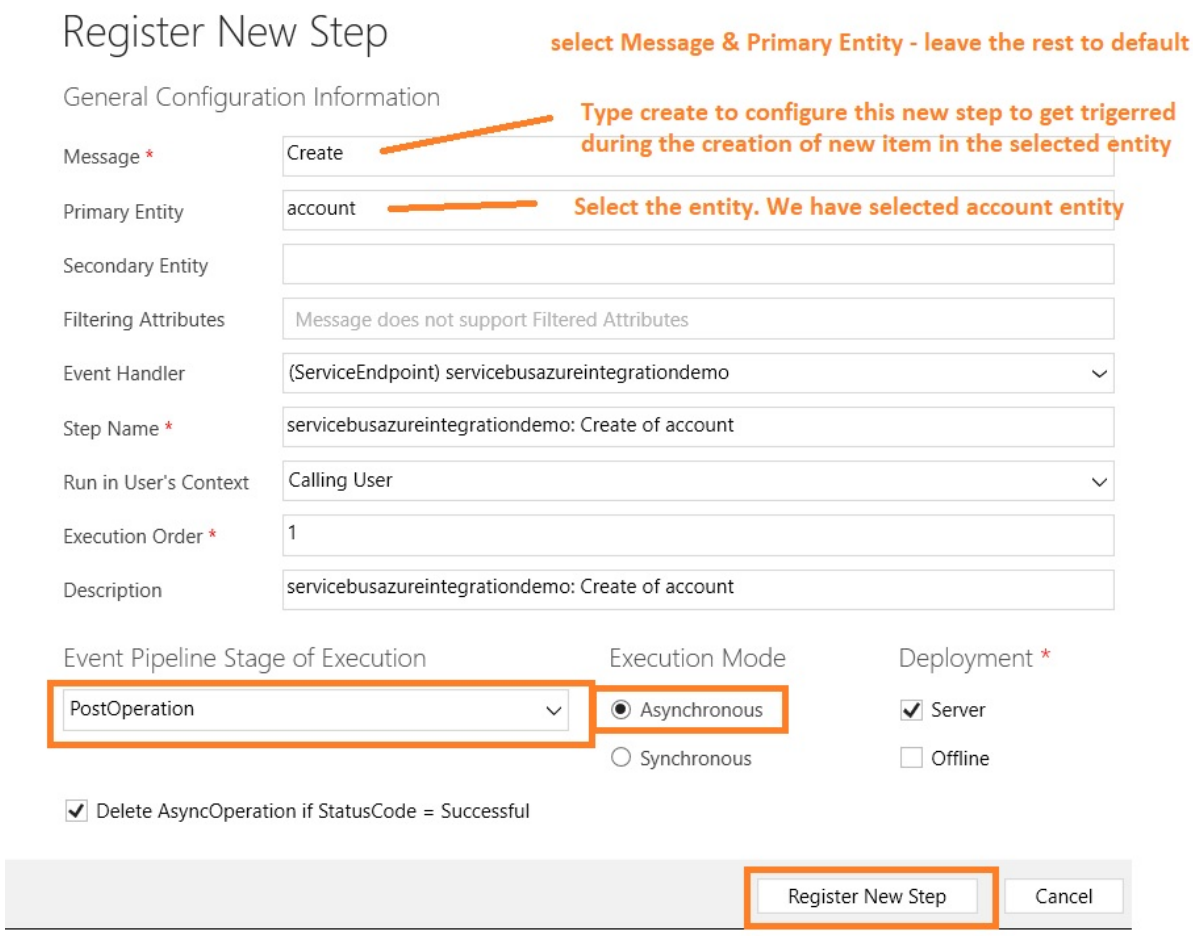


Register New Step for Azure Service Bus service endpoint

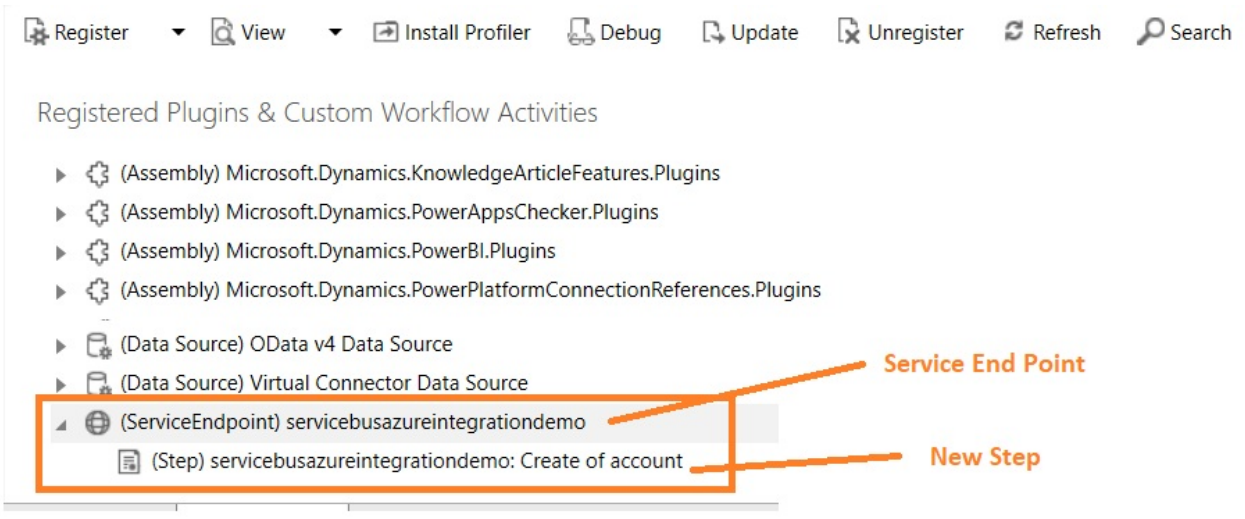
Steps for creating new stept for azure service bus service endpoint in PRT

For creating the Azure aware step, we need to register the New step against the previously created Service Endpoint. Click on the newly created service endpoint, right click and select Register New Step.

In the Register New Step popup , fillin the Message as "Create", Primatry entity as "account" and leave the rest as default and click on Register New Step



Once the Step is created, we can see that both service end point and step are updated in the environment.
Now the configuration part is done, we can go ahead and create new items in account entity in model driven app so that the azure service bus queue will get new json queue message which will contains the details of the newly created account item.

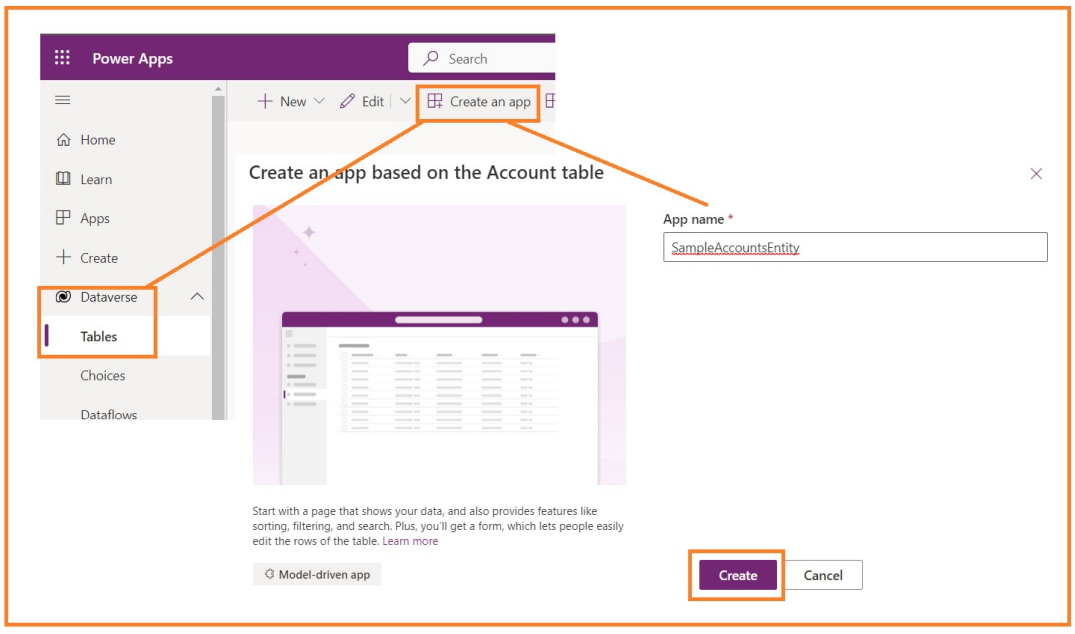


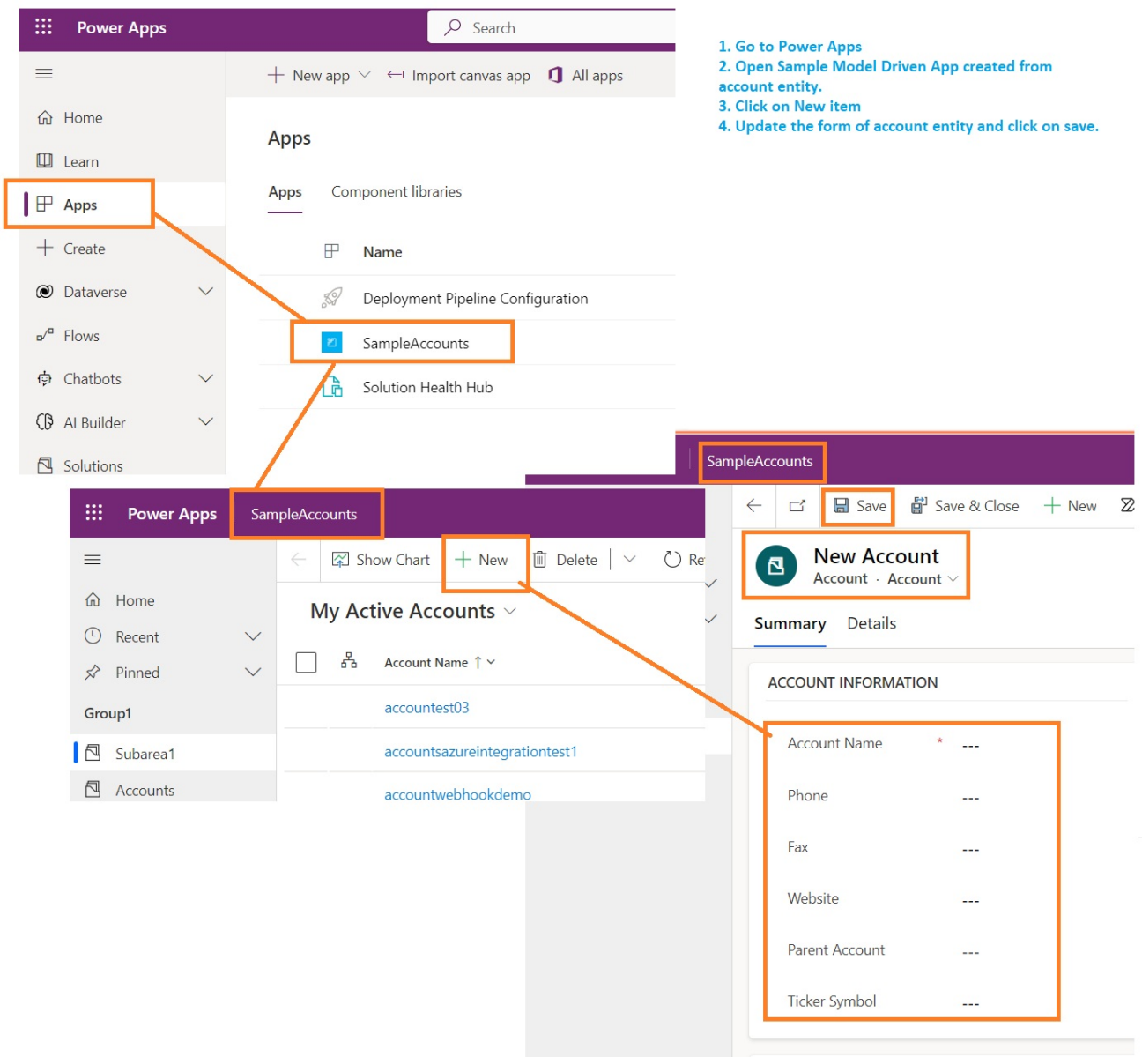
Create a test model driven app based on Account table and test the service endpoint

Steps to trigger the action to azure service bus endpoint for new item

Create a test model drive app linked with account entity or use any existing model driven app linked with account entity. Open the sample Model Driven app, create a new item for account entity and save the item.

Once the item is saved, the azure service bus queue service endpoint will get trigerred and the new item will be sent as a JSON queue message.





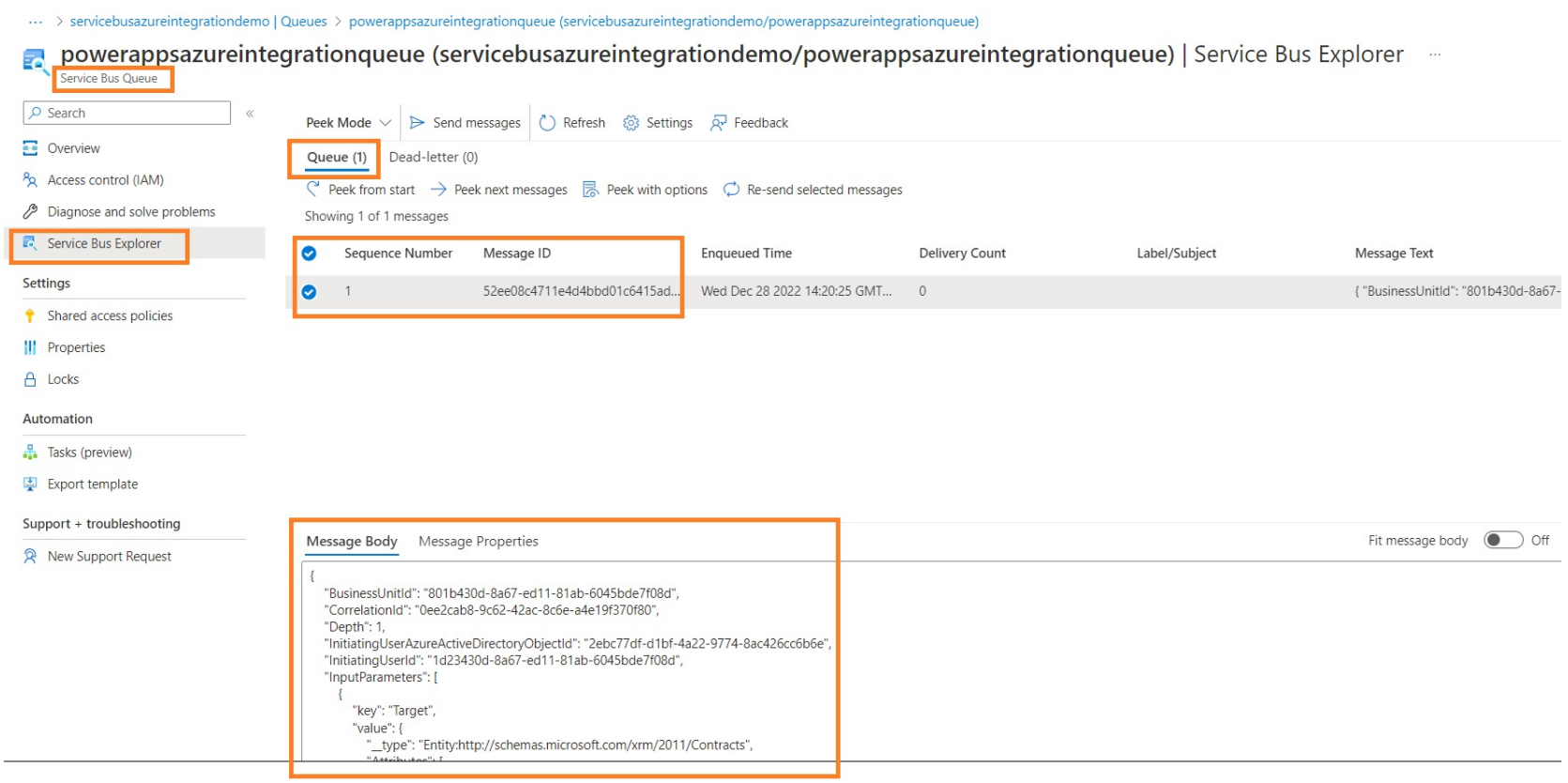
Verify the action results of new item details in Azure Service Bus Queue

Steps to verify the Json response in azure service bus explorer

Once the new account item is created, go to service bus queue inside the azure portal.

Select Service Bus Explorer --> We can see that 1 queue message is avaliable for us to peek.
Peek the message and we can able to see the JSON Queue response message of the newly created account item.

Go to Azure Service Bus queue --> under Service Bus explorer



Once the message is sent to Azure Service bus Queue,

Third Party services listens to the Azure Service Bus queue will peek the message and perform the business scenario.

For this demo, we will create and show Azure Function Service Bus Queue and Power Automate as listeners. Whenever the message gets in to azure service bus queue these listeners will peek these messages and perform the business scenarios.

The listner part is just to show case the full scenario of azure integration with power apps, we will show the listener demo only for azure service bus queue.

For Event hub and Webhook we will create and verify the service endpoint & step.

Create Azure Function Service Bus Queue listener

Steps to create Azure function service bus queue listner to listen / peek new messages from azure service bus queue

Go to Azure Portal.
Search and Create Function app.
Once the function app is created --> click on functions --> create
In create functions --> select Azure service bus queue azure function
FIll the details of the function name and Service bus queue
Click create.
(Refer the below screenshots)

Home > Function App >

Create Function App

Subscription * ⓘ — Free Trial

Resource Group * ⓘ — PowerPlatformDemo

Create new

Instance Details

Function App name * — .azurewebsites.net

Publish * — Code / Docker Container

Runtime stack * — .NET

Version * — 6

Region * — Central US

Operating system

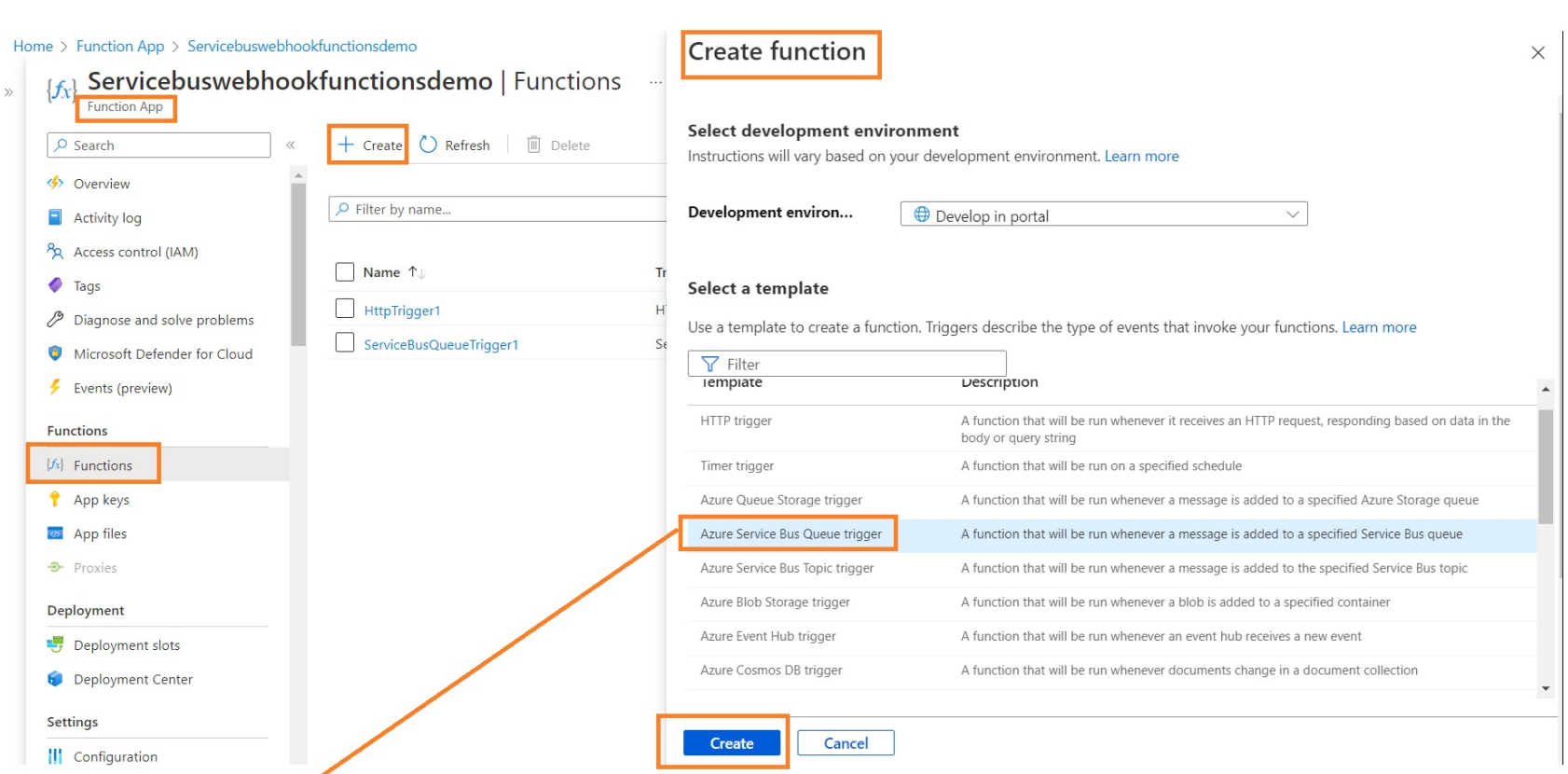
The Operating System has been recommended for you based on your selection of runtime stack.

Operating System * — Linux / Windows

Plan

The plan you choose dictates how your app scales, what features are enabled, and how it is priced. Learn more

Plan type * ⓘ — Consumption (Serverless)



Template details

We need more information to create the Azure Service Bus Queue trigger function. Learn more

New Function *

Service Bus connection * ⓘ

Queue name * ⓘ

Provide the function name — ServiceBusQueueTrigger

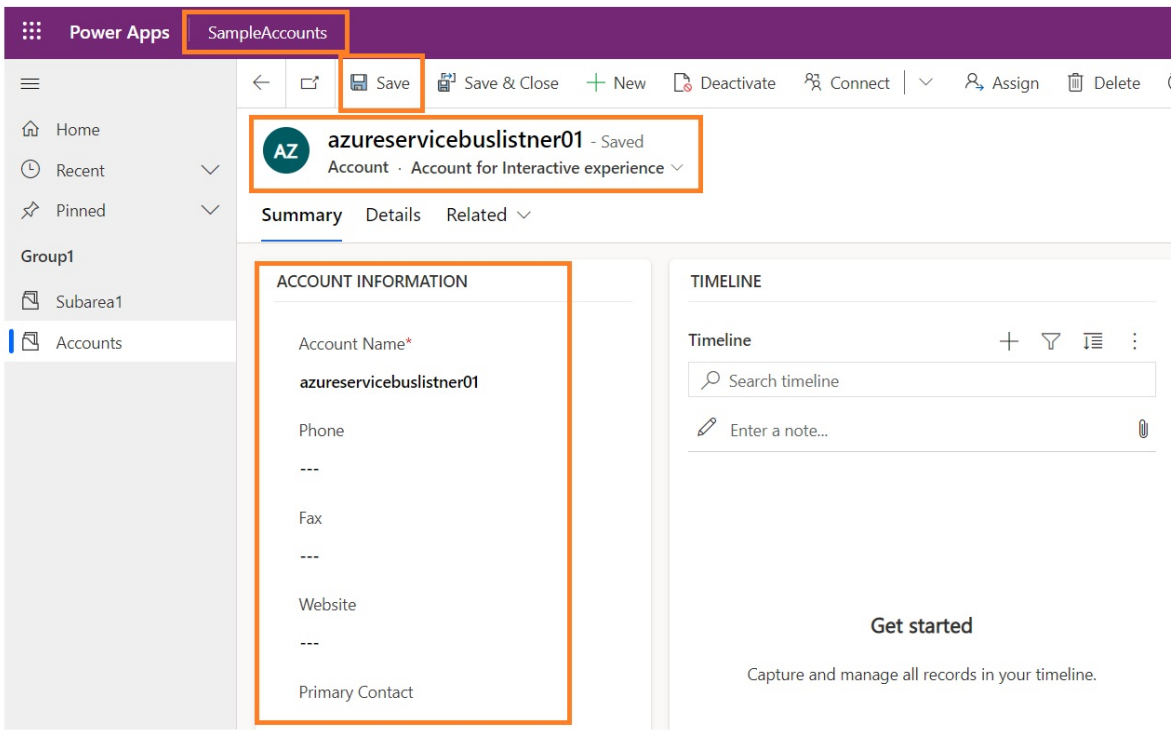
Select the azure service bus queue connection — servicebusazureintegrationdemo_Ro...

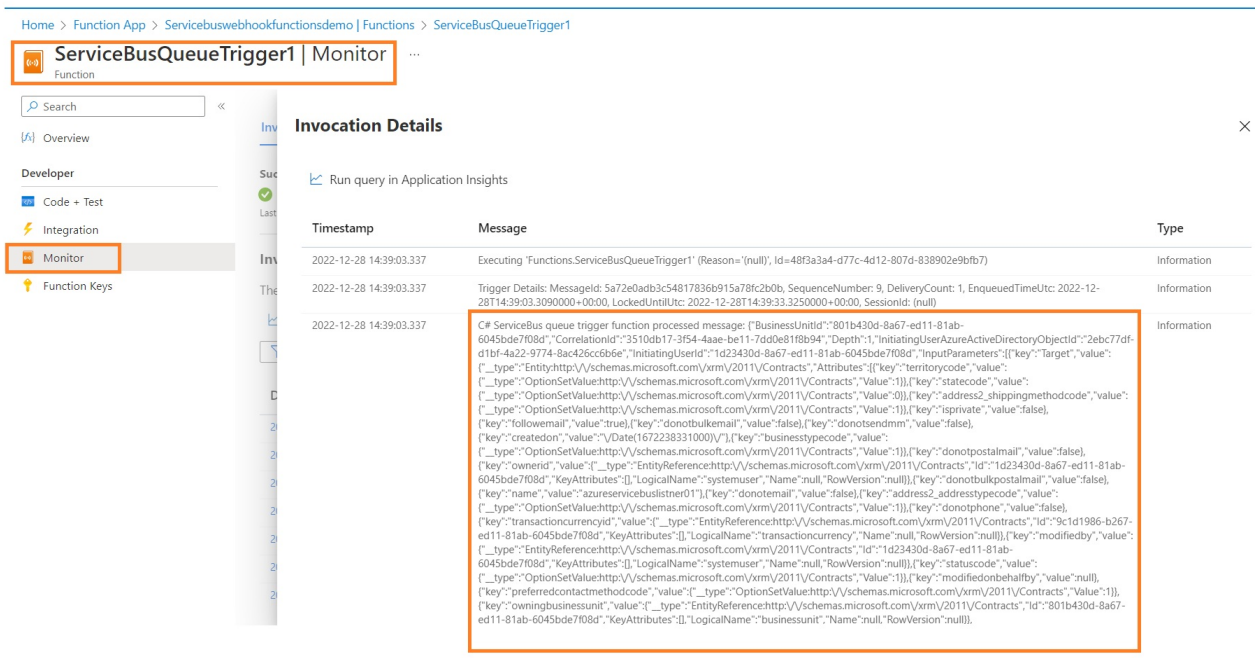
New

Provide the azure service bus queue name created in previous sections — myqueue

Open the sample Model Driven app, create a new item for account entity and save the item.

Once the item is saved, the azure service bus queue will get trigerred and the new item will be sent as a JSON queue message. once the item reaches azure service bus queue, the azure function listener service bus queue will peek the new message.
we can verify this by going to azure portal --> go to azure service bus queue listener function --> under monitor we can see the new json details





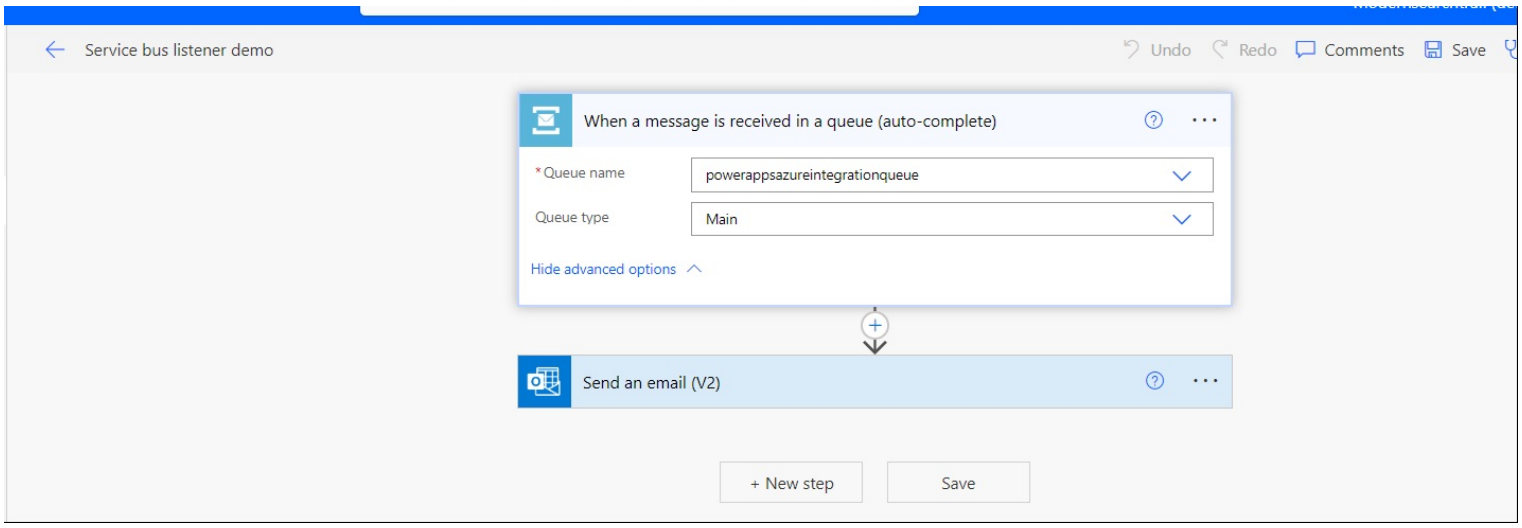
Create Power Automate as Service Bus Queue listener

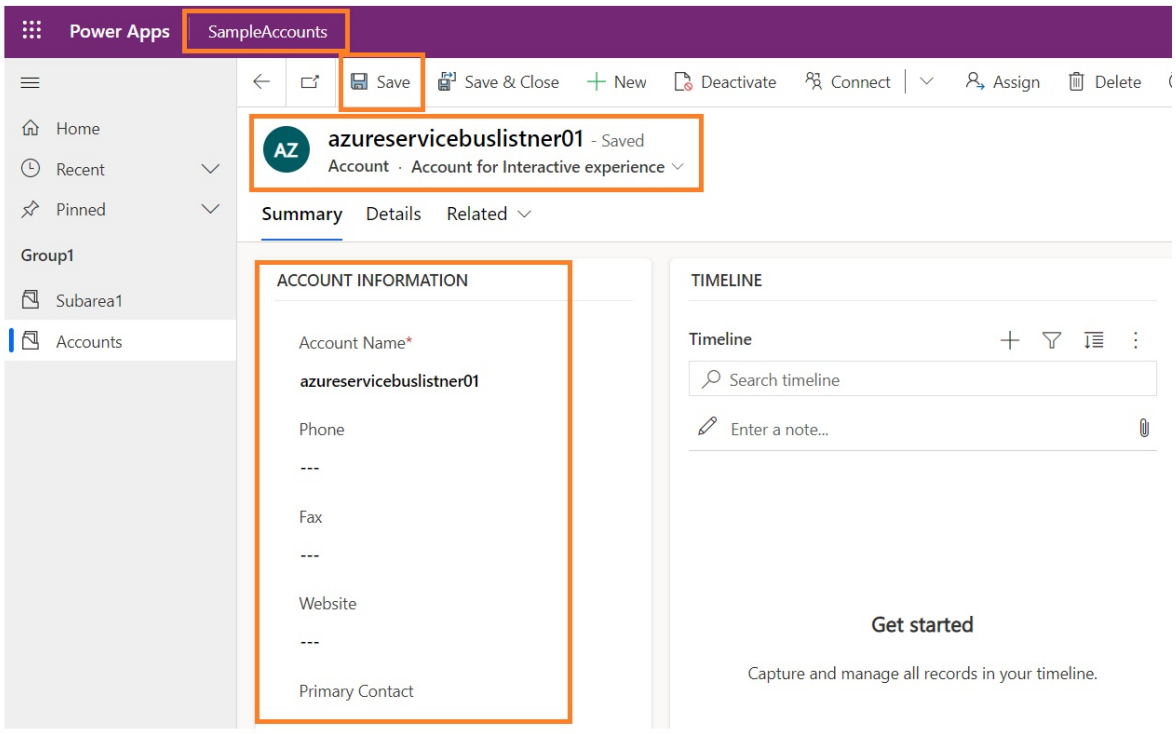
Steps to create Power automate as service bus queue listener to listen / peek new messages from azure service bus queue

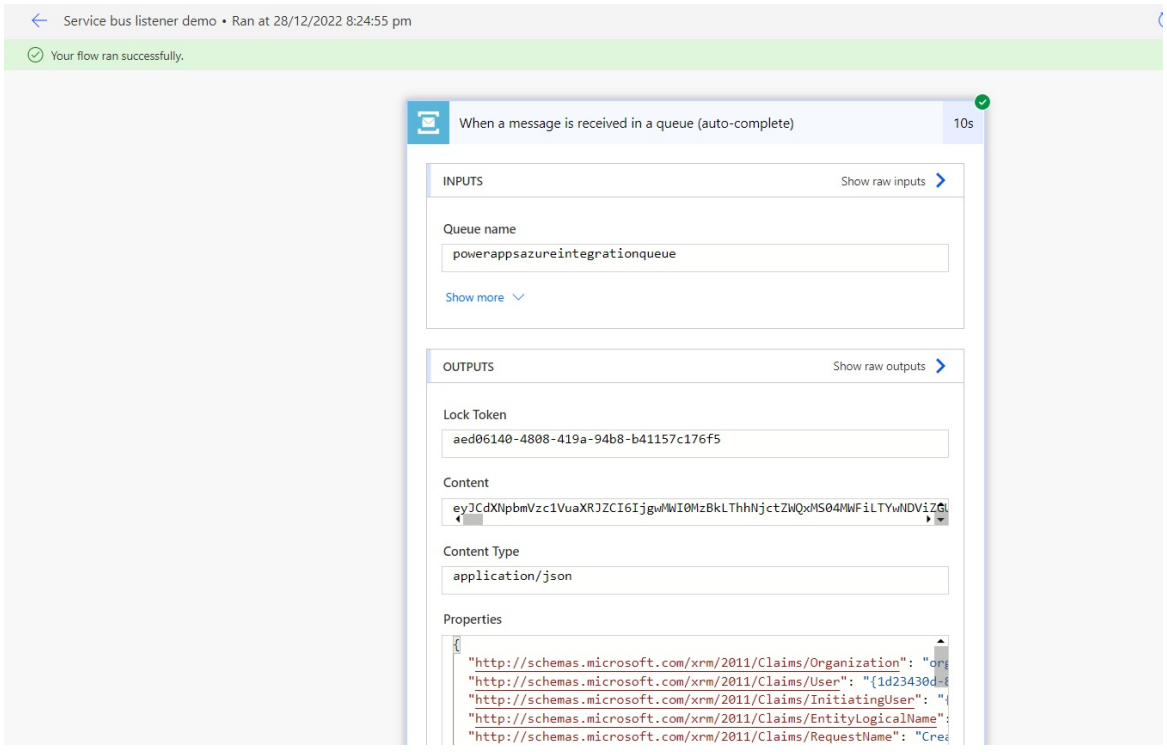
Go to Power Automate --> Create new cloud flow and select when a message is received in a queue as a trigger.
Add the business scenario or some steps and save the flow.

Go to Power APPS --> Open the sample Model Driven app, create a new item for account entity and save the item.

Once the item is saved, the azure service bus queue will get trigerred and the new item will be sent as a JSON queue message. once the item reaches azure service bus queue, the power automate listener service bus queue will peek the new message.
we can verify this by going to Power Automate --> runhistory --> we can see the new json details
(Refer the below screenshots)







This ends the demo of azure service bus. In the next section we will see how to integrate with Azure Event hubs.

Intergrate with Azure Event Hub

Sections:

- **Creation of Azure Event Hub in Azure Portal**
- **New Registration of Service Endpoint for Event Hub**
- **New Step Registration aganist Azure Event Hub Service Endpoint**
- **Trigger the action using the model-driven app**
- **Verify the action message in Azure Event Hub Stream Analytics**

Azure Event Hub Creation in Azure Portal

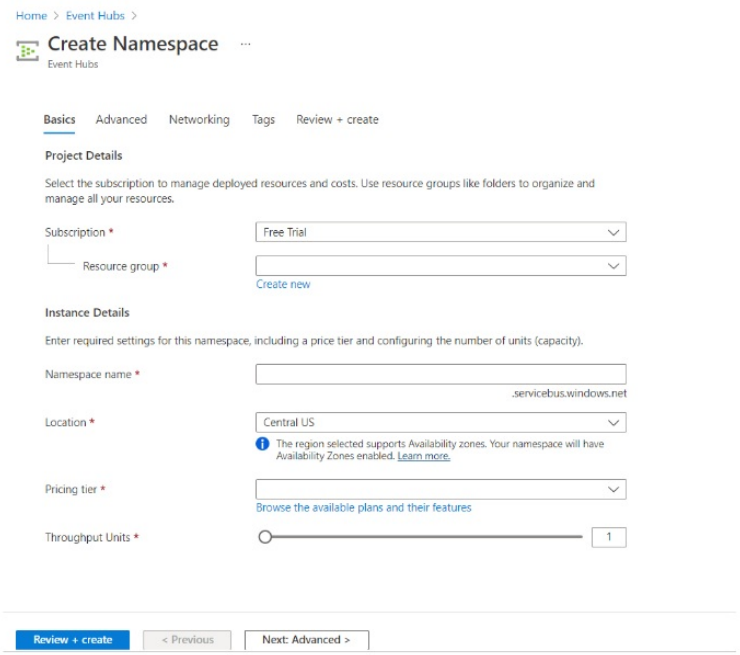
Steps for Creating Azure Event Hub

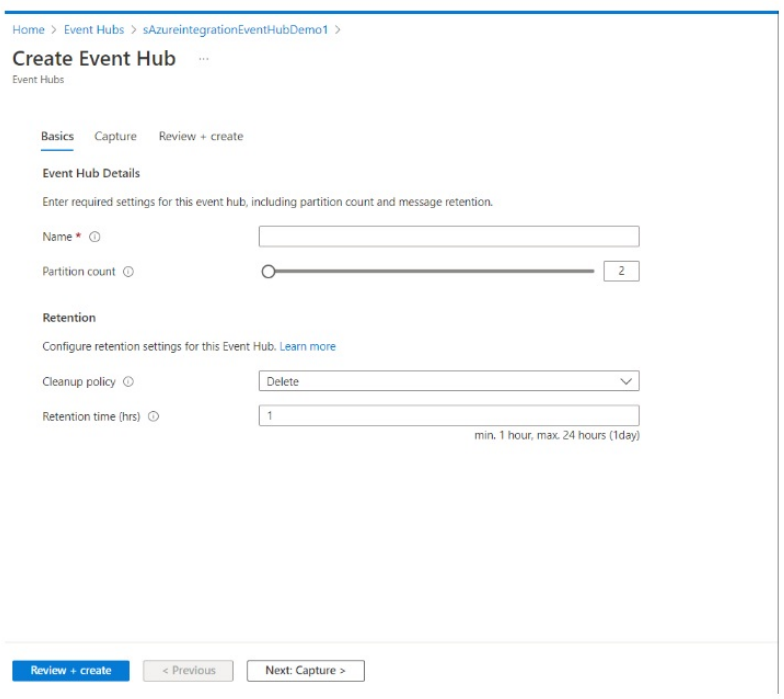
Go to Azure portal and search for Azure Event Hub..
Select Event Hub --> Click on Create NameSpace.
Provide the subscription , resource group, Namespace name, Location, Pricing tier details and click on create.

Once AzureEvent Hub Name space is created, go to Event Hub NameSpace.

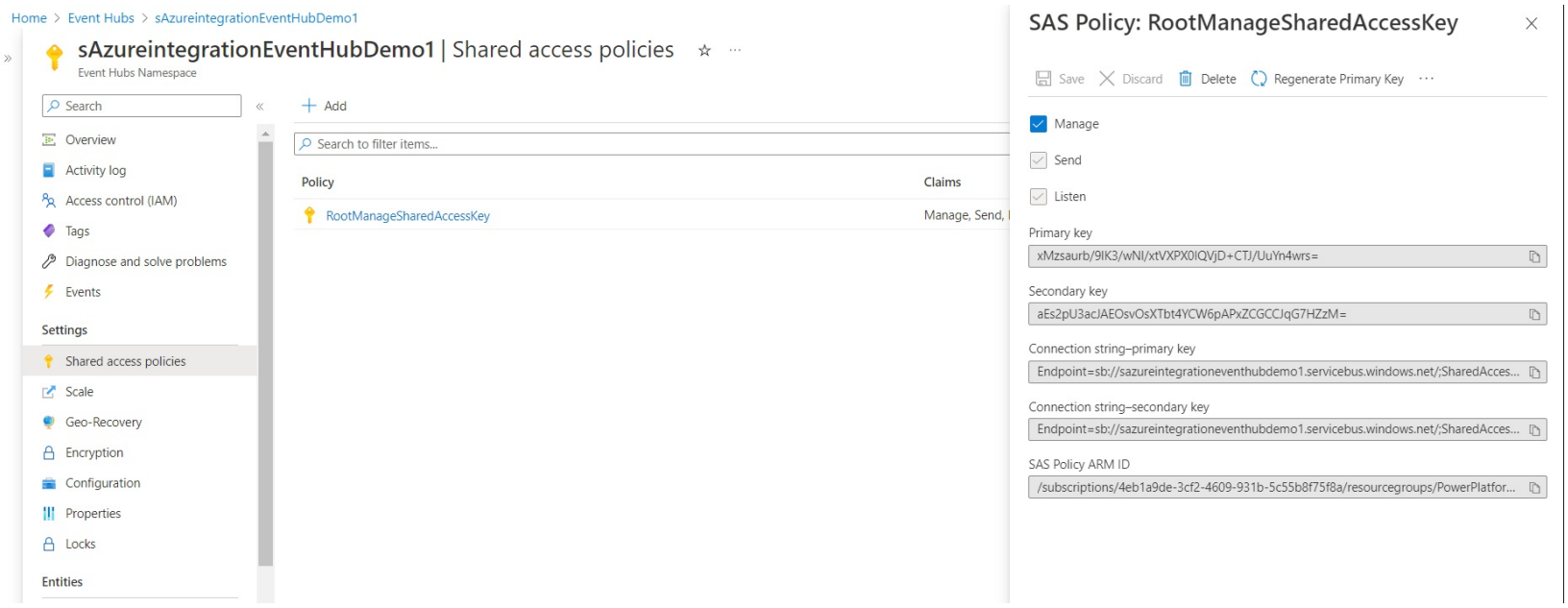
For the demo, we will be creating the Event Hub . So whenever some one creates a new item in the account entity this Event Hub will receive the json message about the newly created account item details via service endpoint and step.

Under Entities --> Click on Event Hub
Provide the name, and leave the other settings to default and create.



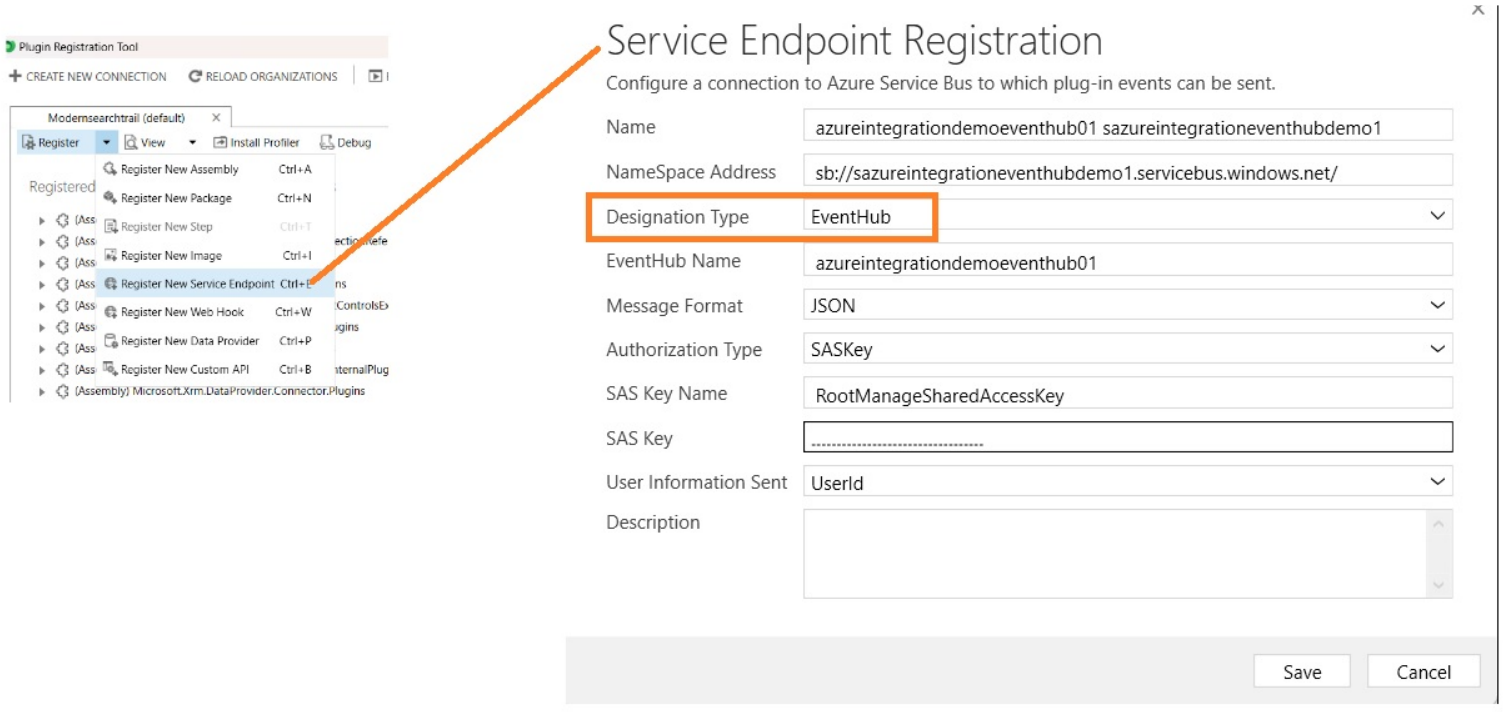


Under Settings --> Click on Shared access policies and copy the connection strings and paste it in notepad. These connection string will be used while creating the service end point in Plugin Registration tool.

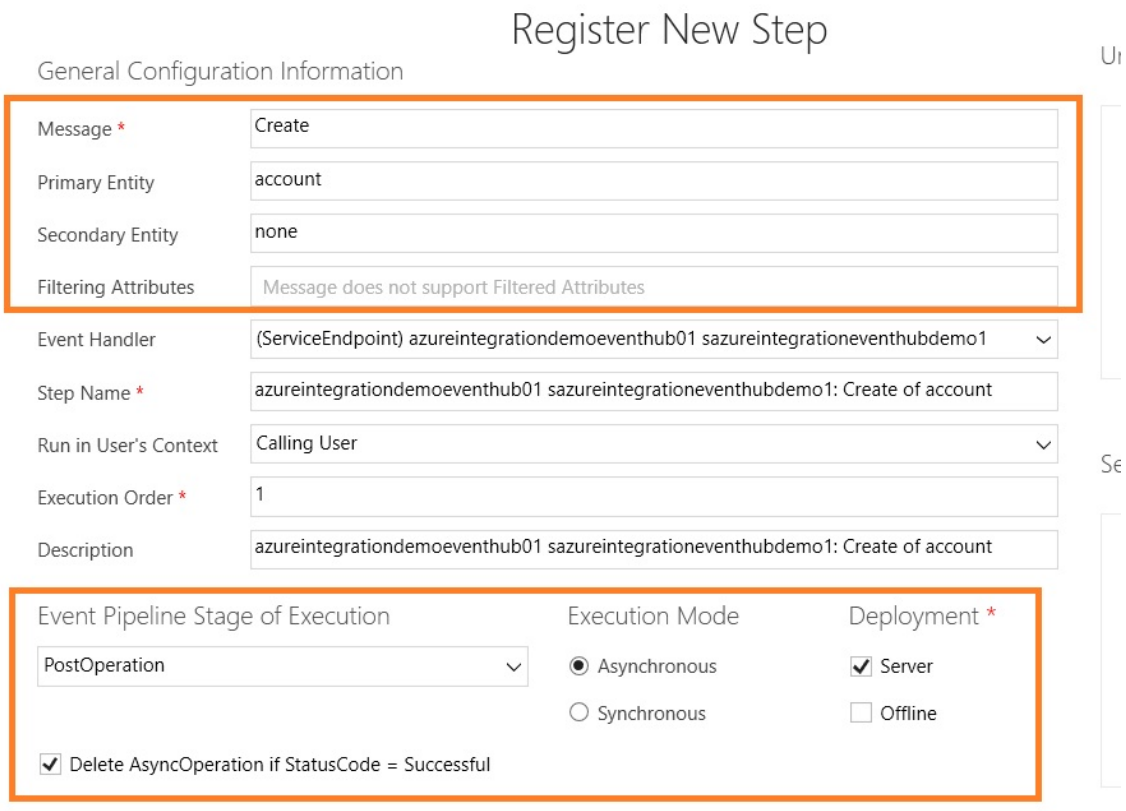


Open PRT tool
Clcik create new connection and login .
Once the New Connection is registered aganist the enviornment.
Select Register --> Click on Register New Service Endpoint
In the Service Endpoint PopUp --> copy the connection string of azure Event Hub copied previously and click on Next.

This will open up the Service Endpoint Registration.
In the Serice Endpoint Registration popup, PRT will pick most of the details and update it to respective section based on the connection string. we need to select the destination type as "EventHub" & Message format as "JSON" & verify the other fields.
Once verified click on Save.
This will create the Azure Event Hub service endpoint registration for this environment.
Next we need Create new step against this service endpoint.
The New Step will be same as the one we configured earlier for service bus. (Refer the below screenshot and create the step again the event hub)

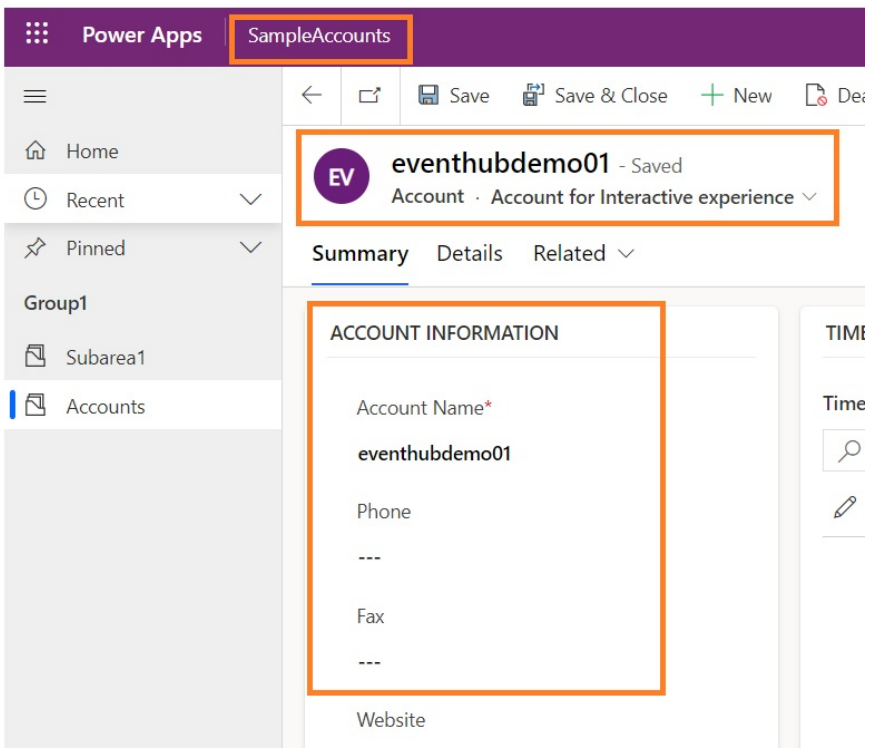


Click on Azure Event Hub Service Endpoint --> right click and select register new step and create the step.



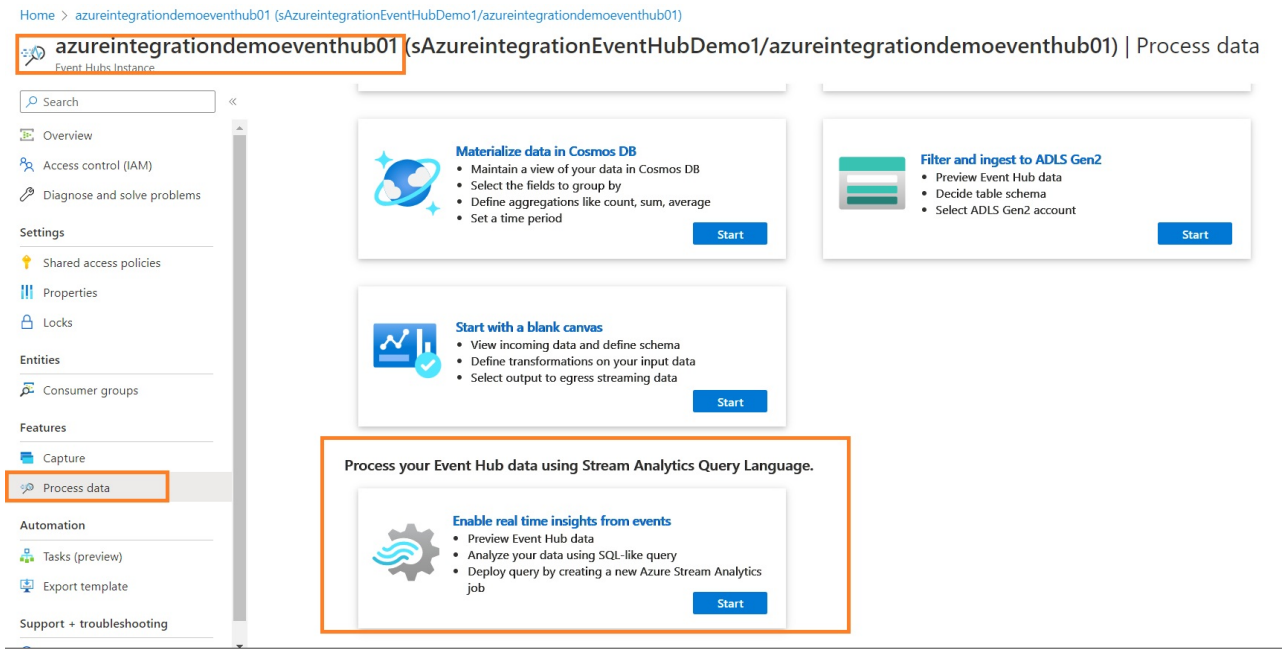
Open the sample Model Driven app, create a new item for account entity and save the item.

Once the item is saved, the azure event hub service endpoint will get trigerred and the new item will be sent as a JSON message.

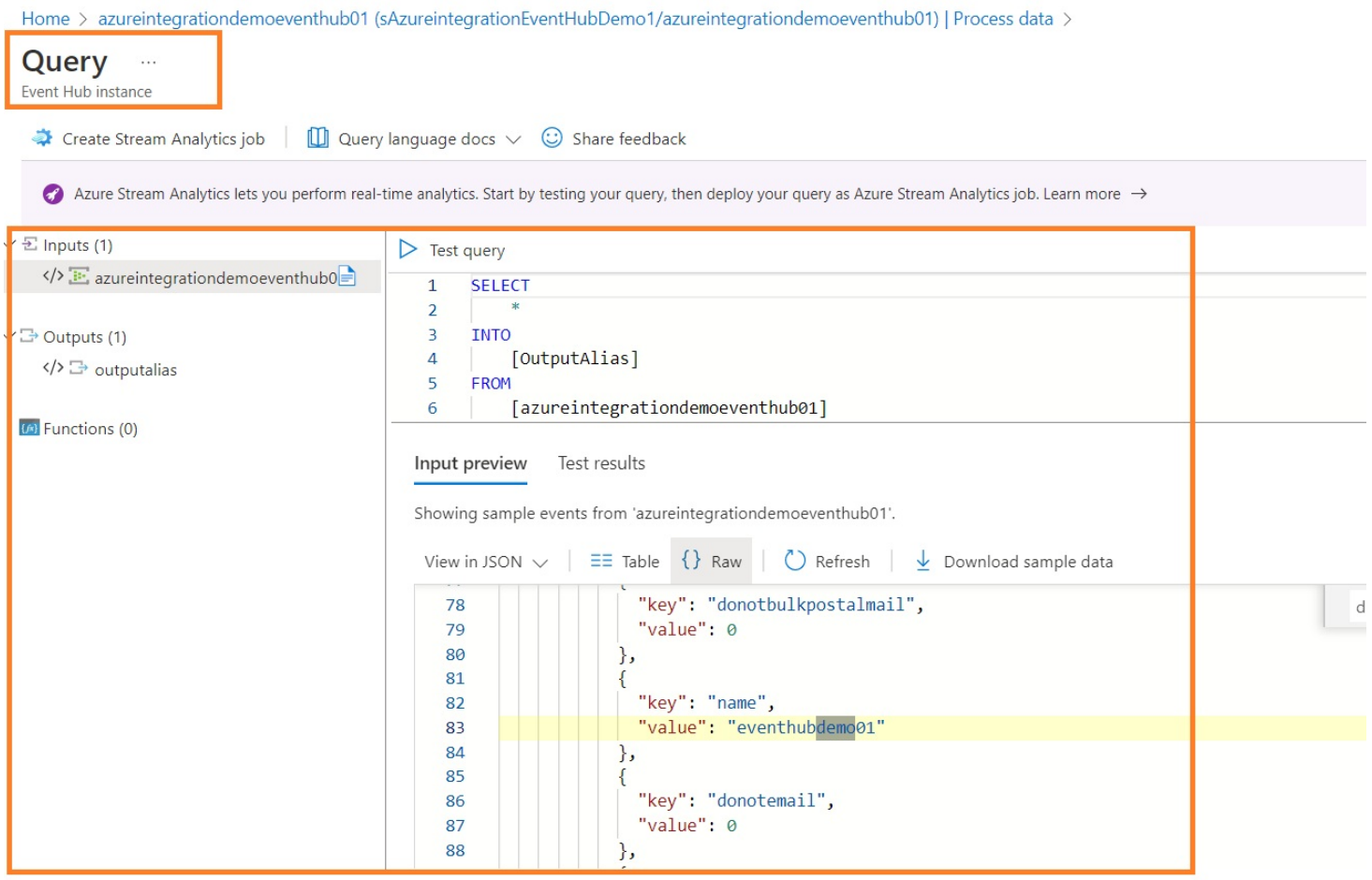


We can verify the new json message response via Azure event hub stream analytics.

Go to azure portal --> Search and select the event hub
Under the event hub --> click on Process data under features.
Click on start under Enable real time insights from events (Stream Analytics Query language)



This will oepn stream analytics query instance.
Select the query and click test query.
we can see the new json response under the preview.



This ends the demo of integration with Azure Event Hub. In the next section we will see how to integrate with webhook.

Intergrate with Webhook

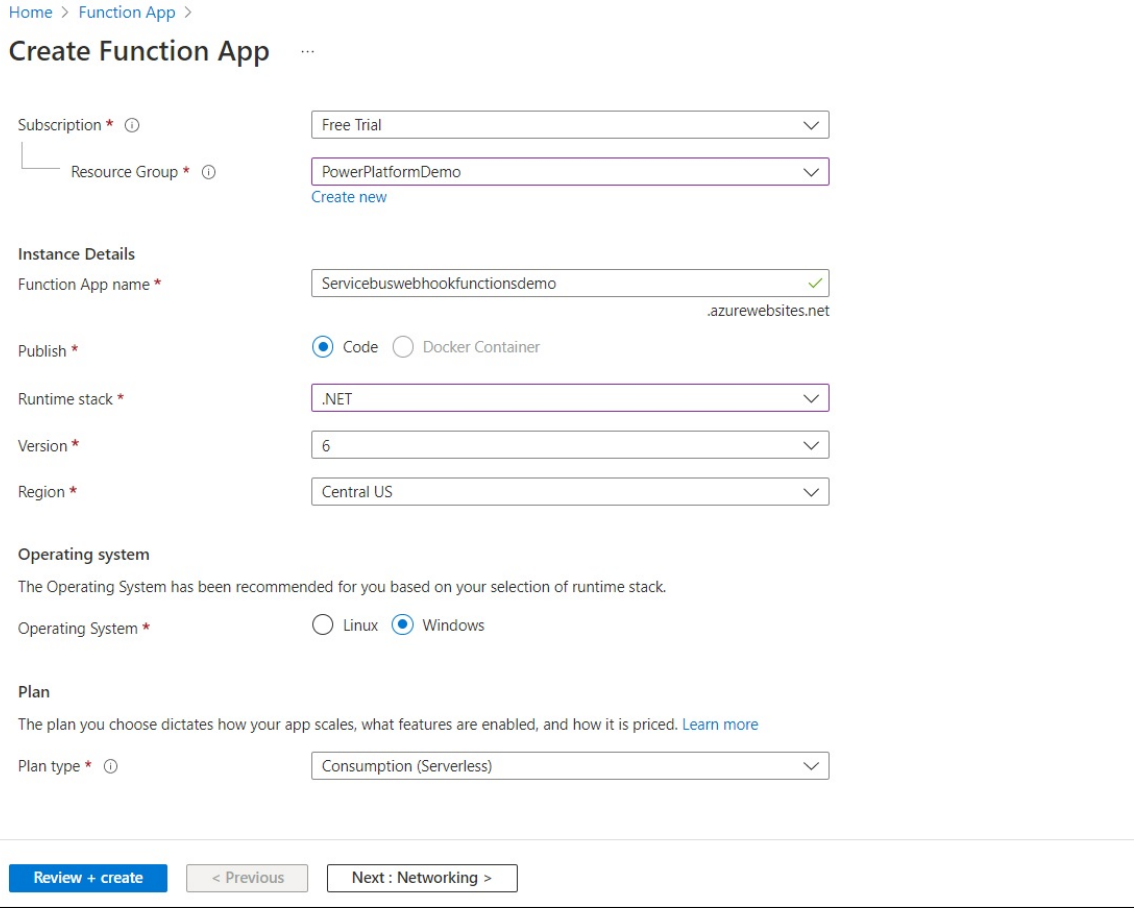
Sections:

- **Creation of Azure Function as http trigerred webhook in Azure Portal**
- **New Registration of webhook Endpoint for azure function**
- **New Step Registration aganist webhook Endpoint**
- **Trigger the action using the model-driven app**
- **Verify the action message in Azure function http trigerred**

Azure Function App Creation in Azure Portal

Steps for Creating Http Trigerred Azure Function as Webhook

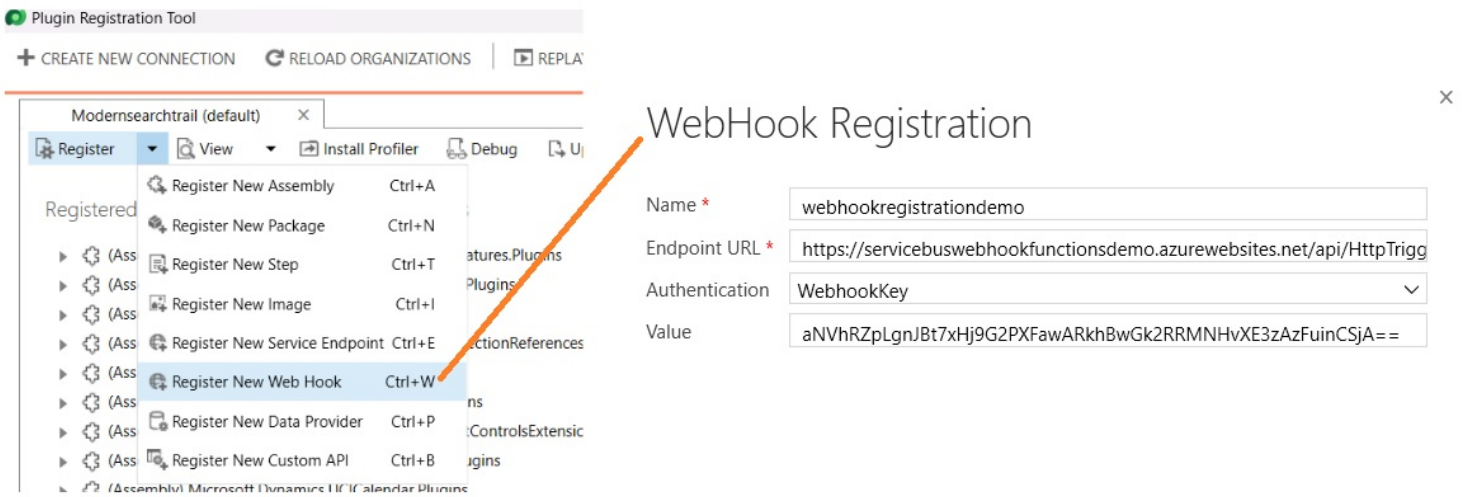
Go to Azure Portal --> search azure function
Create Function App
Create Http Trigerred Azure Function and copy the function url and copy it in notepad

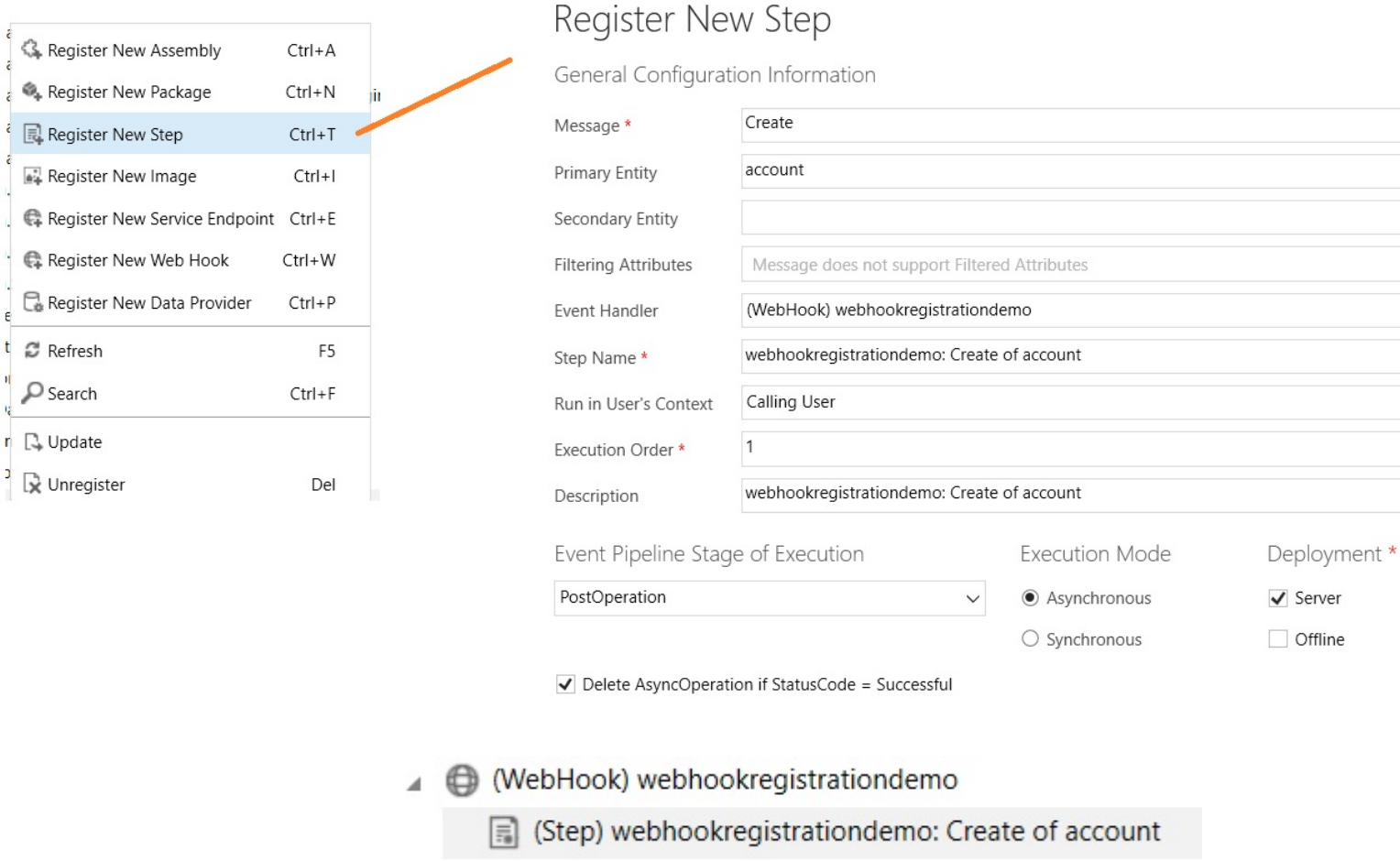


Webhook Endpoint and Step Creation for webhook

Steps for Webhook Endpoint

Open PRT tool
Clcik create new connection and login .
Once the New Connection is registered aganist the enviornment.
Select Register --> Click on Register New WebHook Registration
In the Webhook registration PopUp --> Paste the endpoint url as azure function url and choose authentication as WebhookKey and paste the value of azure function code value copied previously and click on Next.
Once verified click on Save.
This will create the WebHook endpoint registration for this environment.
Next we need Create new step against this webhook endpoint.
The New Step will be same as the one we configured earlier for service bus. (Refer the below screenshot and create the step again the webhook)

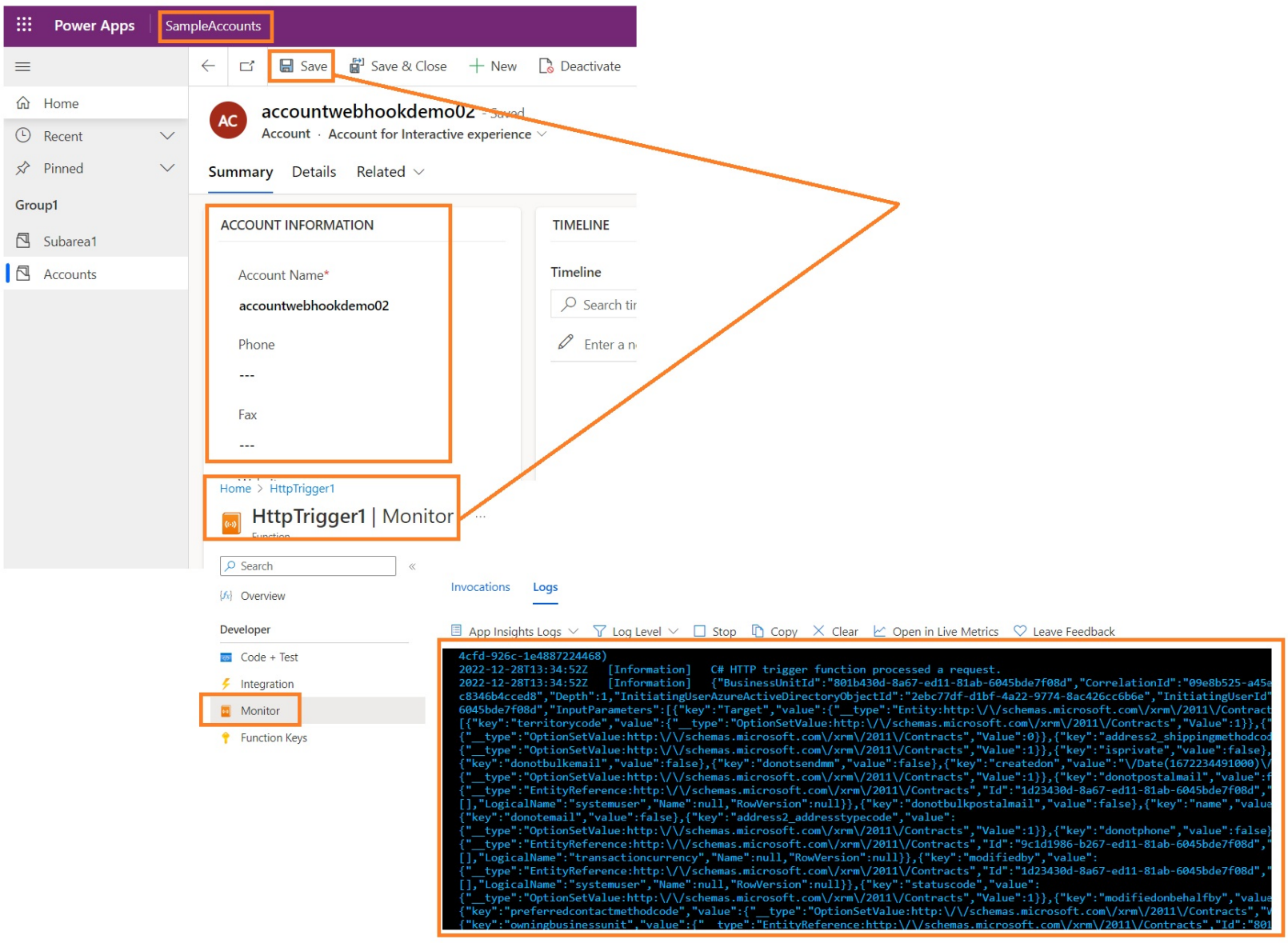




Go to Power APPs
Open the sample model driven app
Add new item for account entity and click on save.

Once saved, the webhook endpoint will get trigerred and the newly added item details will be sent as json to Http Trigerred azure function.

Go to Azure portal
Open Function App
Under monitor --> we can see the new item details.



This is a high level reference document to get started with Azure integration with Power Apps.

Please refer to the below video for the more details regarding the same.

https://youtu.be/-FkIzLHQFfo
https://youtu.be/IwYf9sQ59c8