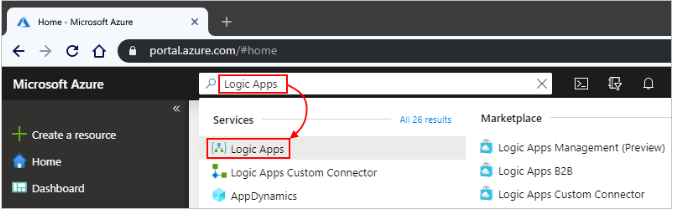
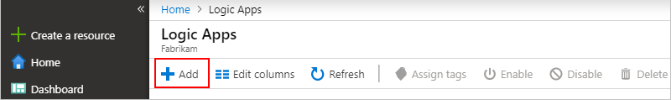
**Logic Apps (Receives all messages from service bus)**

**Creating Logic App that sends email to registered user, when service bus publisher sends message in Azure Portal**

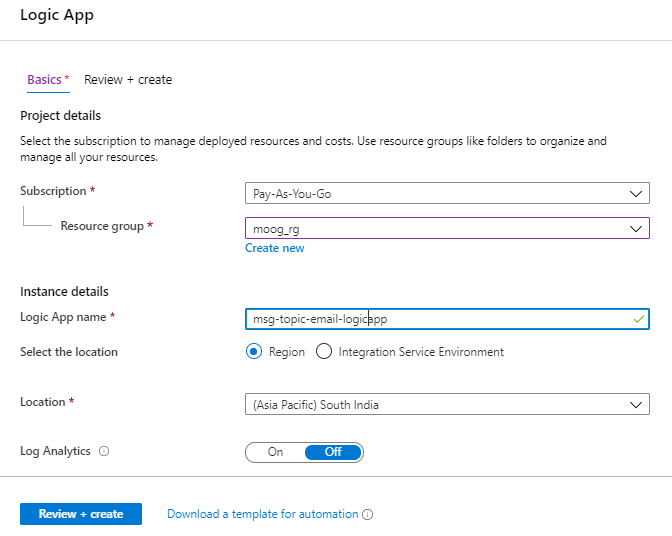
1. Sign in to the Azure portal with your Azure account credentials.
2. From the Azure home page, in the search box, find and select Logic Apps.



1. On the Logic Apps page, select Add.



1. On the Logic App pane, provide details about your logic app as shown below. After you're done, select Create



**Name**: Your logic app name, which can contain only letters, numbers, hyphens, underscores, parentheses, and periods. This example uses "My-First-Logic-App".

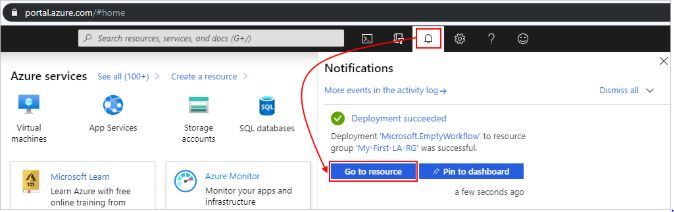
**Subscription:** your azure subscription name.

**Resource group:** The name for the Azure resource group used to organize related resources if created else create new by selecting create new radio button and enter the name.

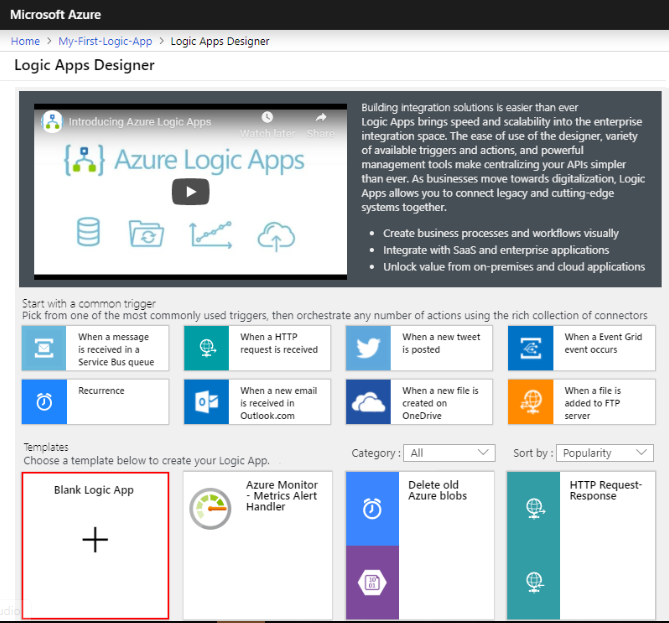
**Location:** The region where to store your logic app information.

**Log Analytics:** Keep the Off setting for diagnostic logging.

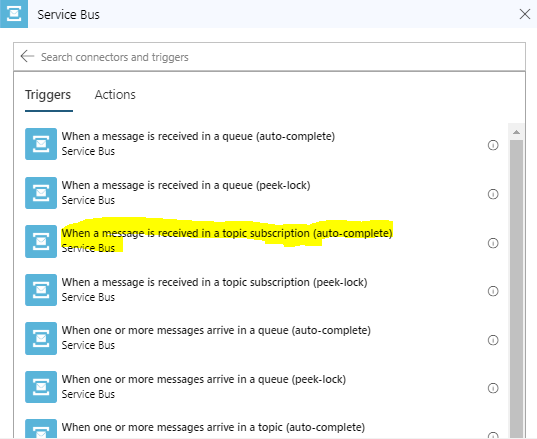
1. After Azure deploys your app, on the Azure toolbar, select Notifications > Go to resource for your deployed logic app or you can find and select your logic app by typing the name in the search box.



1. The Logic Apps Designer opens and shows a page with an introduction video and commonly used triggers. Under Templates, select Blank Logic App.

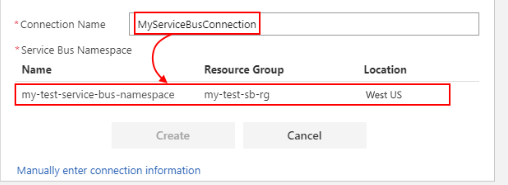


1. After Logic App Designer opens, In the search box, enter "azure service bus" as your filter. From the triggers list, select “When a message is received in a Topic (auto-complete)” trigger that you want.

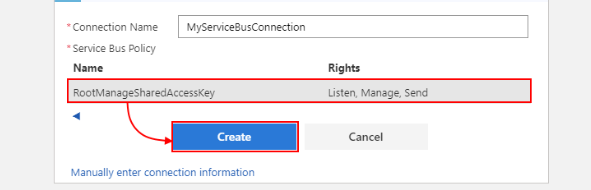


1. If your trigger is connecting to your Service Bus namespace for the first time, follow these steps when the Logic App Designer prompts you for connection information.

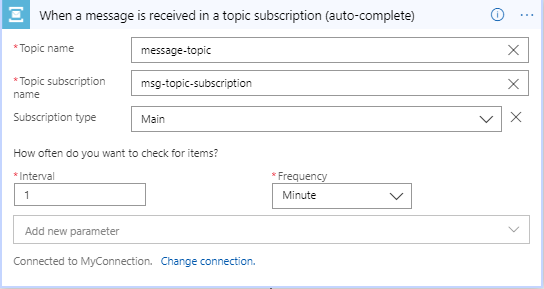
* Provide a name for your connection, and select your Service Bus namespace.

To manually enter the connection string instead, select **manually enter connection information**

9. Select your Service Bus policy, and select **Create**.

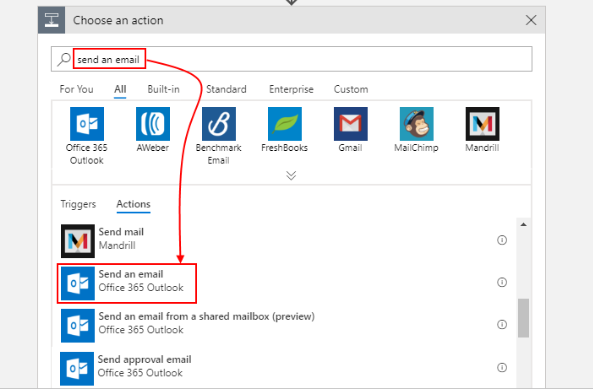


10. Enter Topic name and subscription create in azure portal as shown below



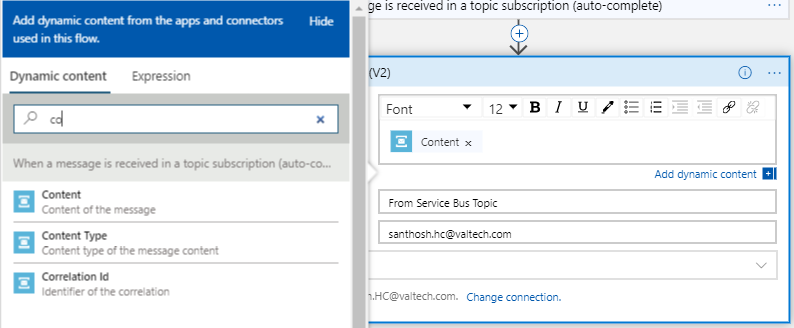
11. Then click on New Step button to create action to send email.

12. Search for send an email connector and select as shown below

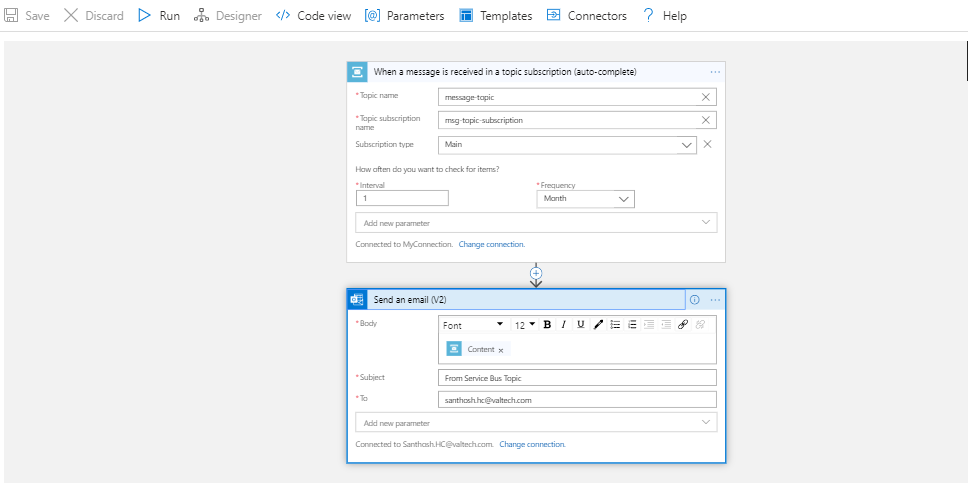


13. If your selected email connector prompts you to authenticate your identity, complete that step now to create a connection between your logic app and your email service.

14. Enter To and subject of the email, then select content as body for email. Here content selected Dynamic content dialog box will be the message sent by the service bus publisher



15. Save the logic App and app looks as below



16. Once service bus topic publisher sends message, then email will be triggered to “**To”** email address entered in email connector

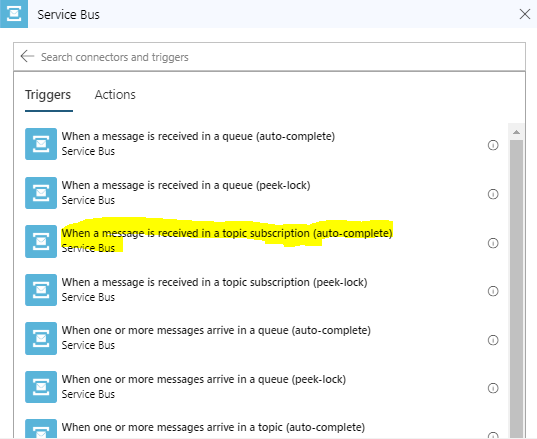
**Creating Logic App that that inserts data into azure table storage, when service bus publisher sends message in Azure Portal.**

## Prerequisites

1. Create Table in azure table storage with proper partition key.

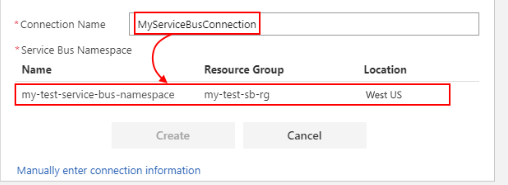
Steps

1. Create blank Logic app, following steps in previous logic app
2. After Logic App Designer opens, In the search box, enter "azure service bus" as your filter. From the triggers list, select “When a message is received in a Topic (auto-complete)” trigger that we want.

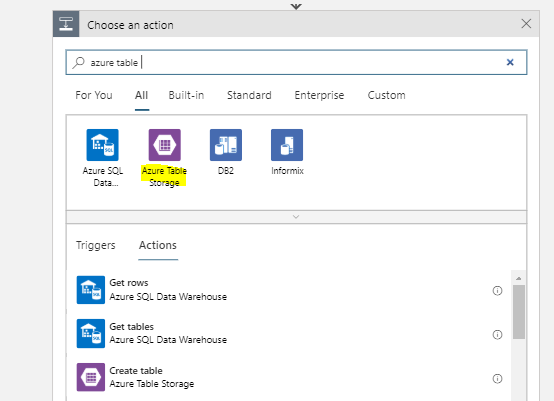


1. If your trigger is connecting to your Service Bus namespace for the first time, follow these steps when the Logic App Designer prompts you for connection information.

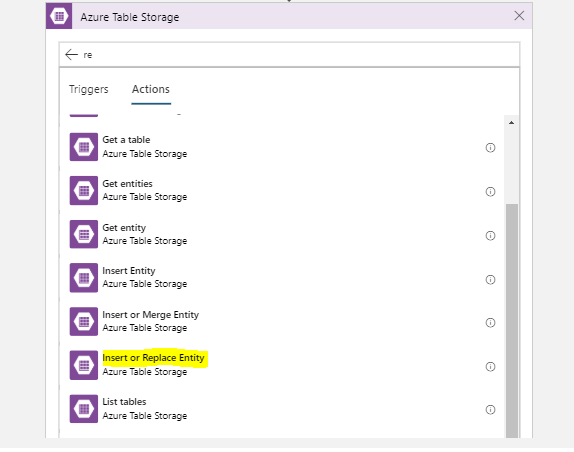
* Provide a name for your connection, and select your Service Bus namespace.

To manually enter the connection string instead, select **manually enter connection information.**

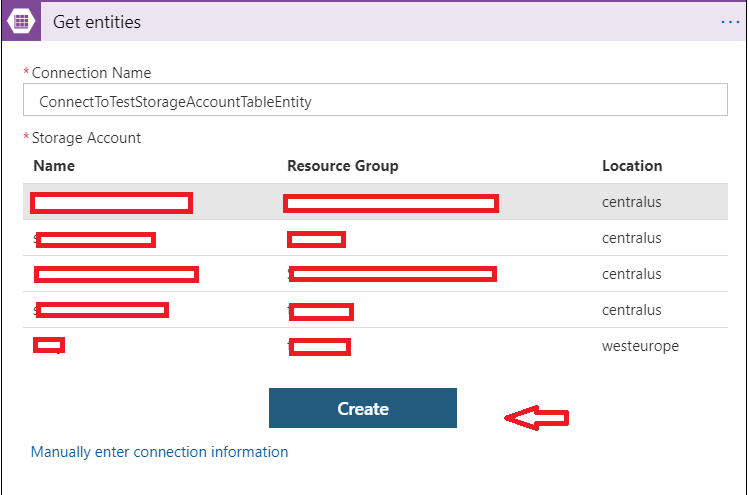
1. Then click on New Step button and search for Azure Table Storage to insert message into table storage.



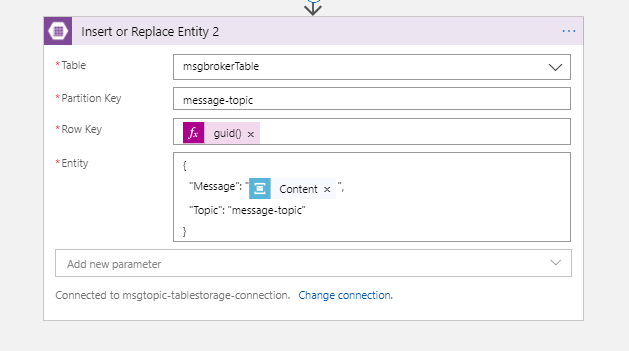
1. To Insert or replace message select “Insert or Replace Entity” from action as below.



1. Now we need to set the connection to access the entities of Storage Table. Provide the connection name and select the intended storage account, click on Create button. After this, the Storage connection context should have been created.



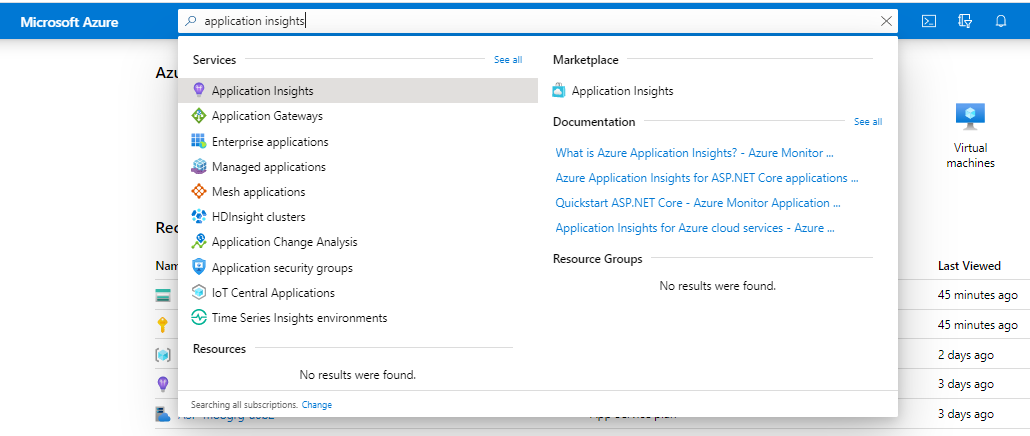
1. After connection configuration, enter Table name , partition key and row key value of the table that created in azure table storage and enter entity details that we need to insert in Entity box as blow.(content in entity box is message sent by publisher)



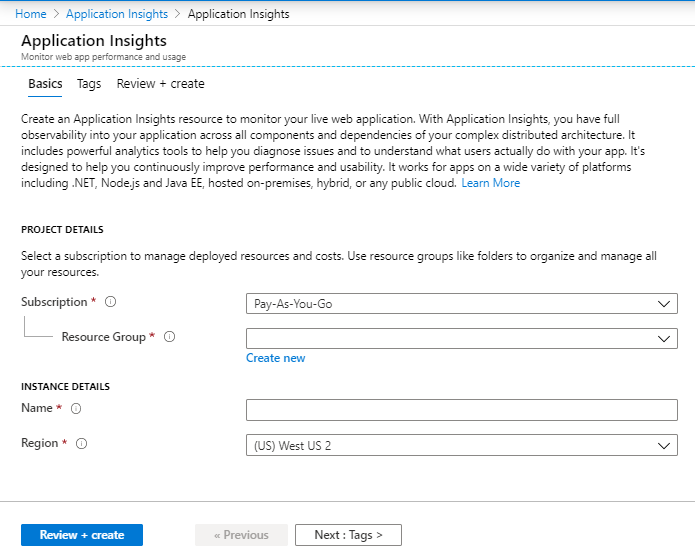
1. When service bus topic publisher sends message this logic app will be triggered and message will be inserted into table

**Create logic app that requests http request to web Api application that logs message in application insights**

1. First, we need to create application insight service in azure portal. Follow below steps to create the same
2. Sign in to the Azure portal, and search for Application Insights and select that

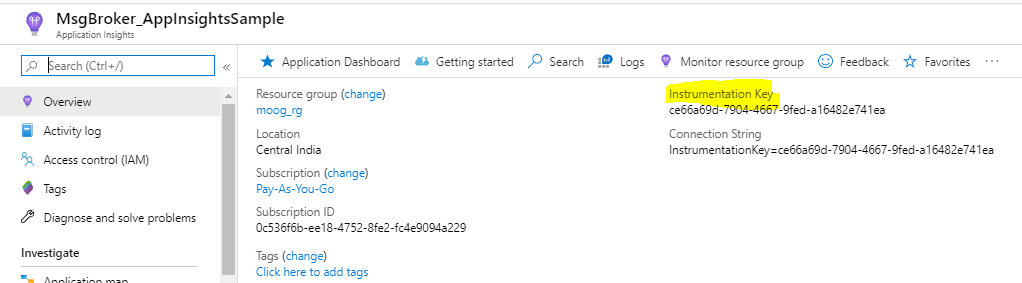


1. Enter the appropriate values into the required fields, and then select **Review + create**.



| **Settings** | **Value** | **Description** |
| --- | --- | --- |
| **Name** | Unique value | Name that identifies the app you are monitoring. |
| **Resource Group** | myResourceGroup | Name for the new or existing resource group to host App Insights data. |
| **Location** | South india | Choose a location near you, or near where your app is hosted. |

1. Copy “Instrumentation Key” from home page of Application Insights overview screen as shown below



1. Create web API application in visual studio and make below changes to log to message to app insights
2. Install Application Insights from Mange nugget package.
3. In **startup** class add below in configure service method

services.AddApplicationInsightsTelemetry();

1. In **appsettings.json** file specifying the **Instrumentation Key** copied from azure portal



1. Enable ILogger traces, set log level and configure with app insight in program.cs file. Make following changes as below

public class Program

{

public static void Main(string[] args)

{

CreateWebHostBuilder(args).Build().Run();

}

public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>

WebHost.CreateDefaultBuilder(args)

.UseApplicationInsights()

.UseStartup<Startup>()

.ConfigureLogging(

builder =>

{

builder.AddApplicationInsights("Instrumentation Key"); //past instrumentation key

builder.AddFilter<Microsoft.Extensions.Logging.ApplicationInsights.ApplicationInsightsLoggerProvider>

("", LogLevel.Information);

}

);

}

1. Add controller any controller and add any action that receives message as a parameter and log that message as below

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Logging;

namespace ServicebusTopic\_AppInsightsLogging.Controllers

{

[Route("api/[controller]/[action]")]

[ApiController]

public class ValuesController : ControllerBase

{

private readonly ILogger \_logger;

public ValuesController(ILogger<ValuesController> logger)

{

\_logger = logger;

}

// GET api/values

// GET api/values/5

[HttpGet]

public ActionResult<string> GetTopicMessage(string message)

{

if (string.IsNullOrEmpty(message))

\_logger.LogWarning("Message cannot be empty");

else

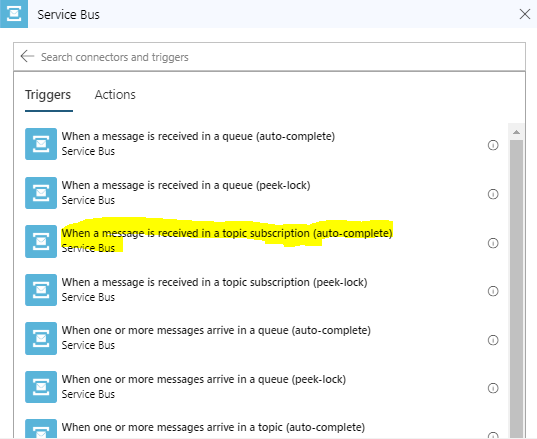
\_logger.LogInformation(message);

return "value";

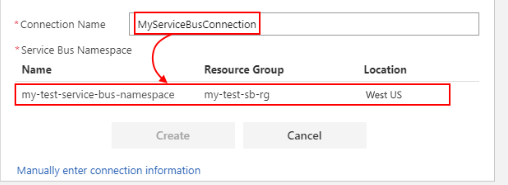
}

}

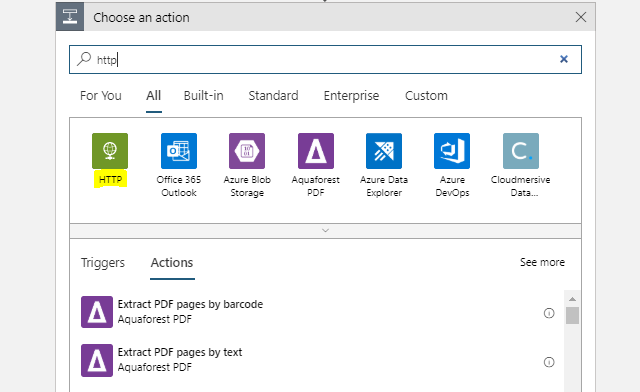
}

1. Now application is ready. So host application azure and copy the url.
2. Create logic app that request triggers http request to log message in app insights.
3. Create blank Logic app, following steps in previous logic app
4. After Logic App Designer opens, in the search box, enter "azure service bus" as your filter. From the triggers list, select “When a message is received in a Topic (auto-complete)” trigger that we want.
5. If your trigger is connecting to your Service Bus namespace for the first time, follow these steps when the Logic App Designer prompts you for connection information.

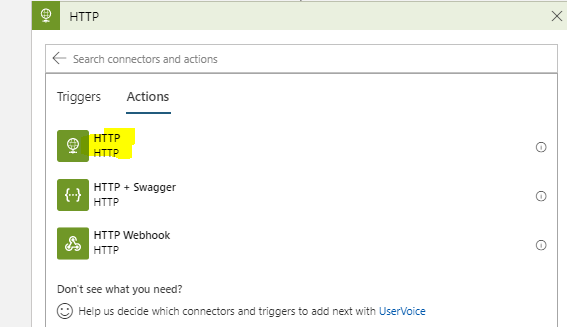
* Provide a name for your connection, and select your Service Bus namespace.

To manually enter the connection string instead, select **manually enter connection information.**

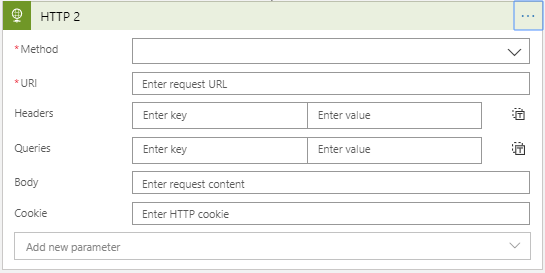
1. Then click on New Step button and search for Http to insert message into App insights.



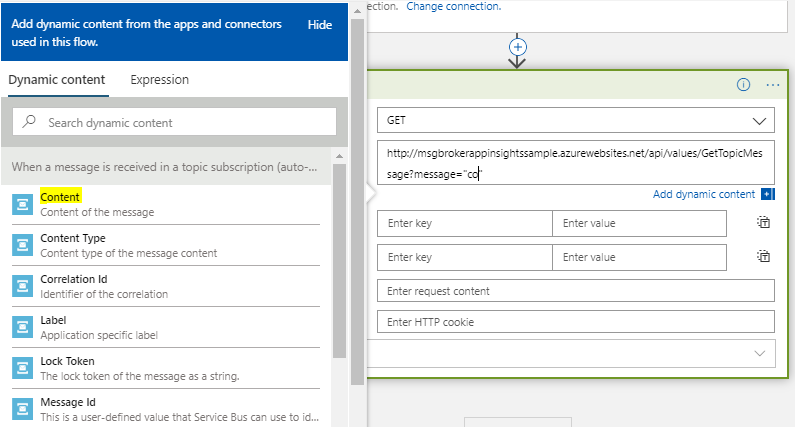
1. Select Http in Action list.



1. Enter appropriate details below



**URL** : The url that requests action method that logs message in app insights. Pass message in query sting as below



Select content form dynamic express dialog for getting message received by service bus connector**.**

1. Now save logic app. So that our logic app is ready to log message in app sights.

