Microsoft Copilot Studio

Lab 04: Build and call Power Automate cloud flows

Hands-on lab step-by-step

January 2025

UDPP Copilot Studio Workshop

Contents

Microsoft Copilot Studio

This lab is subject to the Terms of Use found at the end of this document.

Goals for this lab

After this lab you will be able to: • Understand the basics of Power Automate • Use Copilot Studio to request data from another data source using Power Automate in a basic use case (using the ServiceNow connector) and return the data in a conversational dialog with a customer or user	The time to complete this lab is [60] minutes.

Prerequisites

Labs have been designed to be completed with only a Microsoft Copilot Studio trial. You can start most labs without having to complete the previous module but note that some exercises may reference previous labs. To fully experience the features and functionality of the product, it is recommended that you make sure to have completed all pre-requisites below before starting this lab.

For this lab you need:

- A computer with internet access.
- Be able to log into the provided Microsoft tenant (some companies enforce users to only connect to their company tenant) or your own enterprise tenant with a Copilot Studio User License (or trial)
- **Generative AI should be set to "classic" (in Settings, Generative AI)**
- **Access to Power Automate (make.powerautomate.com)**
- **Access to an active ServiceNow instance (URL, login and password) – don't forget to wake the instance if you use a trial**

Fundamental Knowledge: Understanding Power Automate

Power Automate is a cloud-based service that makes it practical and simple for line of business users to build workflows that automate time-consuming business tasks and processes across applications and services.

Power Automate is part of a powerful and adaptable business application platform that includes Power Apps, Microsoft Dataverse, Dynamics 365, and Office 365. This platform allows our customers, our partners, and our ISV partners to create purpose-built solutions for their own companies, their industry, for functional roles or even for specific geographies. Line-of-business users, who understand their business needs best, can now easily analyze, compose, and streamline data and processes. Professional developers can easily extend the automation, analytics and apps line-of-business to leverage Azure services like Functions, App Service, and Logic Apps. API connectors, gateways, and Microsoft Dataverse make it possible to get more value out of services or data already in use, either in the cloud or on-premises.

Here are a few examples of what you can do with Power Automate.

- Automate business processes.
- Send automatic reminders for past due tasks.
- Move business data between systems on a schedule.
- Connect to more than 1500 data sources or any publicly available API.
- You can even automate tasks on your local computer like computing data in Excel.

Microsoft Copilot Studio connects easily with Power Automate, being able to pass the variables from user's responses and retrieve data from several different data sources, perform complex operations on that data and return to Microsoft Copilot Studio to share that data with the user. Being able to operate on and retrieve data from almost any data source accessible via an API is one of the most valuable benefits of Copilot Studio.

Alternatively, Microsoft Copilot Studio can also call the same connectors, HTTP requests, or custom connectors as in Power Automate, directly from a topic or from a plugin action.

As part of this Microsoft Copilot Studio lab, it will not include an extensive introduction to Power Automate but does cover a basic scenario of how you can retrieve data from an external data source and use it in the conversational experience of Copilot Studio. To learn more specifically about Power Automate, you can review the Microsoft Docs on Power Automate and also review the in a day material for Microsoft Power Apps, which includes Power Automate.

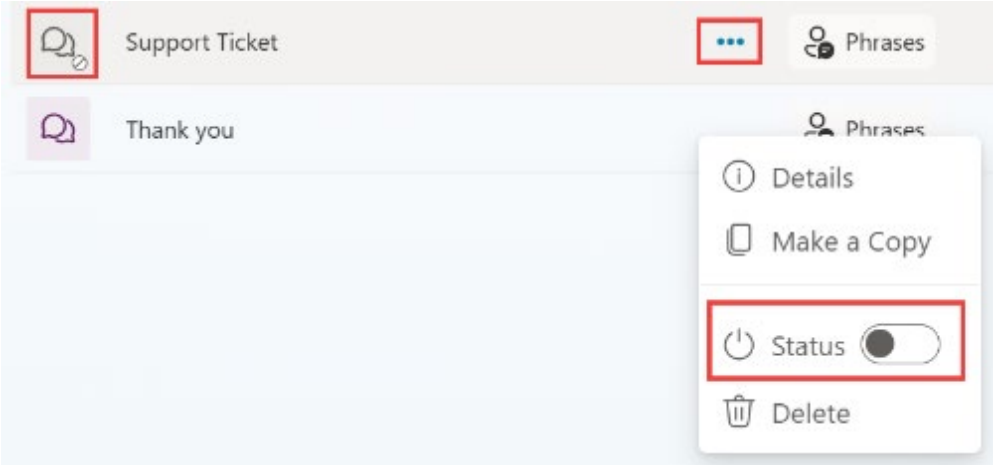
Exercise 1: Build a basic Power Automate cloud flow

Connecting to data provides companies with some of the most benefits as it provides information and insight to users that is up to date and often the relevant for customer or user questions.

In this exercise, you will go through creating a new topic, adding a simple Power Automate action to retrieve information from an external service, and display that data back to the user.

Task 1: Create a new topic

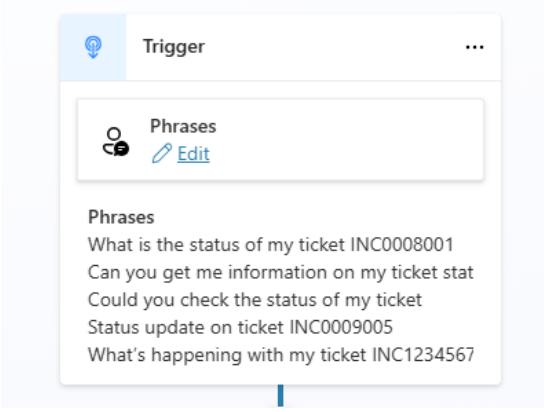
1. To avoid confusion with topics created in previous labs, **disable** any topic related to **Support Ticket**.



2. Open the Copilot Studio authoring canvas' **Topics** page and click **Add a topic** drop down at the top of the screen, select the **From blank** option. Name your topic:
`Check Ticket Status`

3. Add some **trigger phrases** that a user may ask such as the below ones (if you don't see trigger phrases, make sure your "Generative AI" settings are set to "classic")

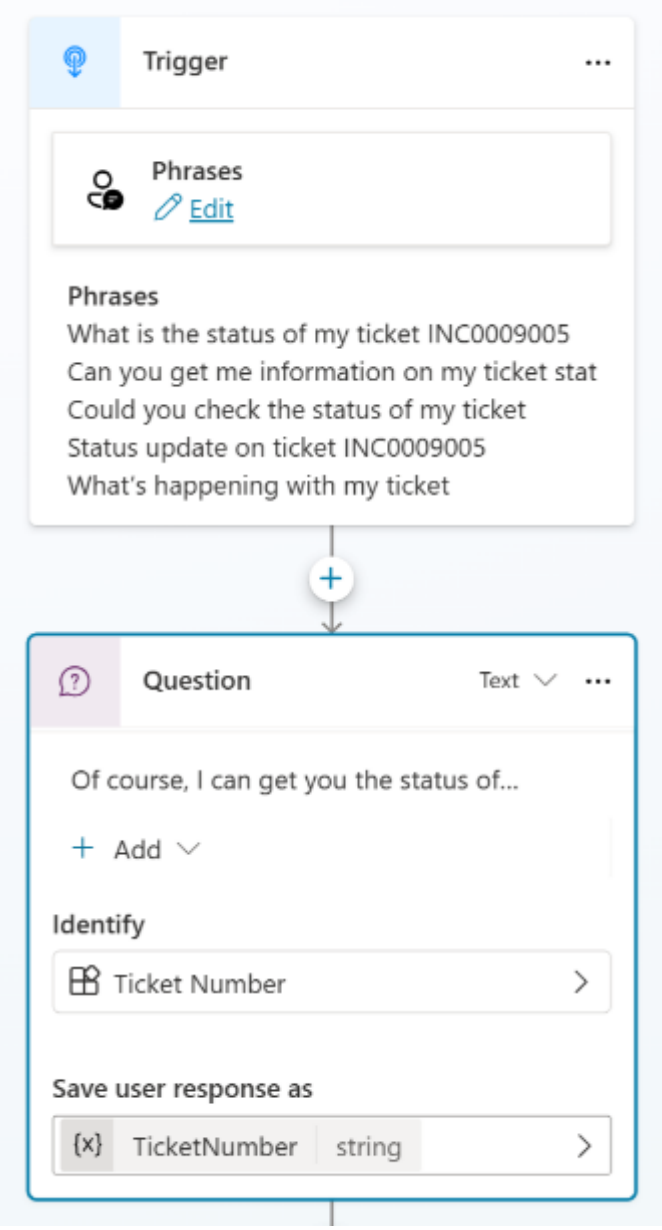
```
What is the status of my ticket INC0008001
Can you get me information on my ticket status
Could you check the status of my ticket
Status update on ticket INC0009005
What's happening with my ticket INC1234567
```



4. Create a new **Question** node and enter text:
`Of course, I can get you the status of your ticket. What is your ticket number?`

5. In **Identify**, choose **Create an Entity** of type **Regular expression (Regex)**, call it `Ticket Number`
and with this pattern:
`INC[0-9]{7}`

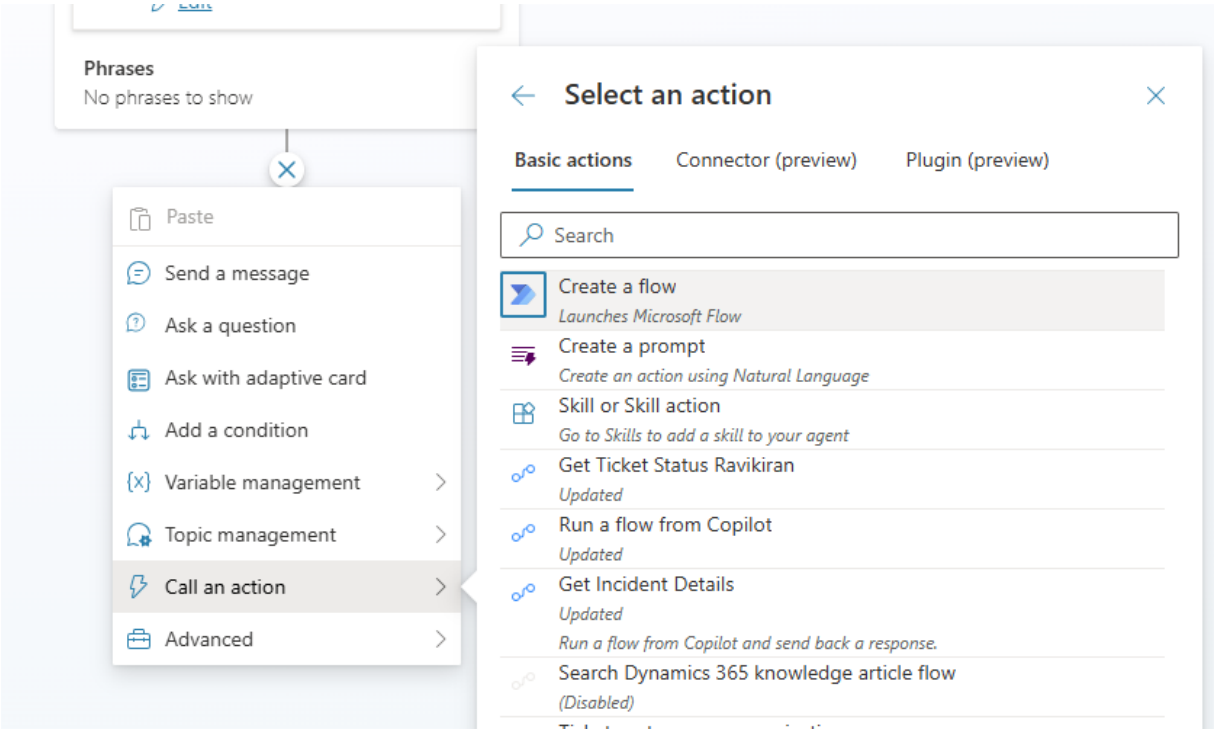
6. Rename the **Var1** variable to `TicketNumber`



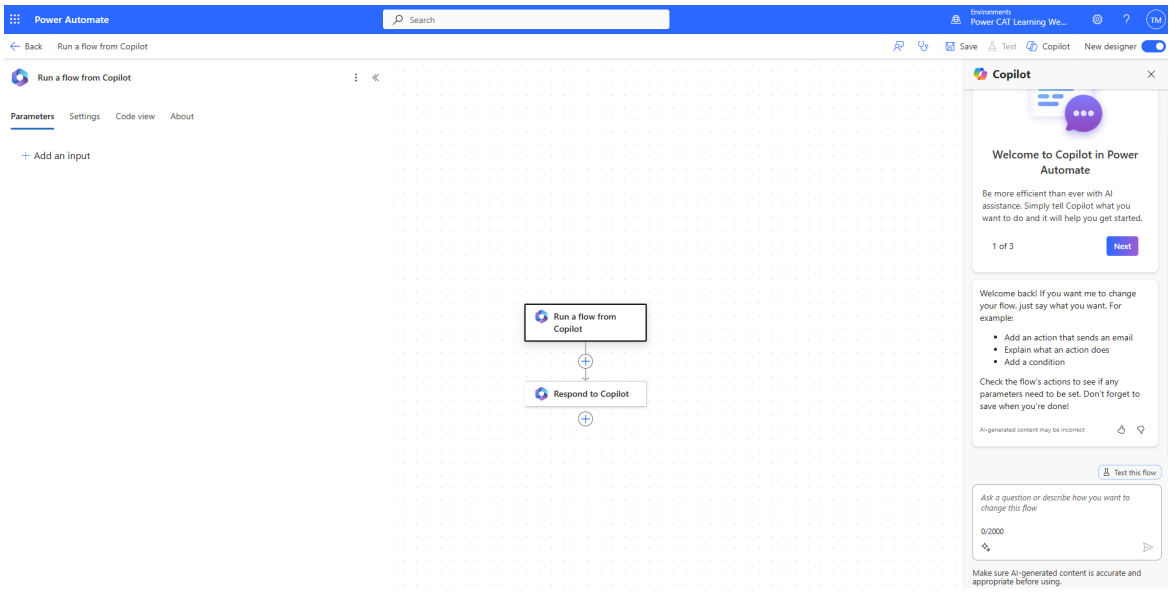
7. **Save** your topic.

Task 2: Create your Power Automate cloud flow

1. Click on the **(+)** new node button under the **Question** node, and select **Call an action** and then click **Create a flow** underneath **Basic actions** as shown in the screenshot below.

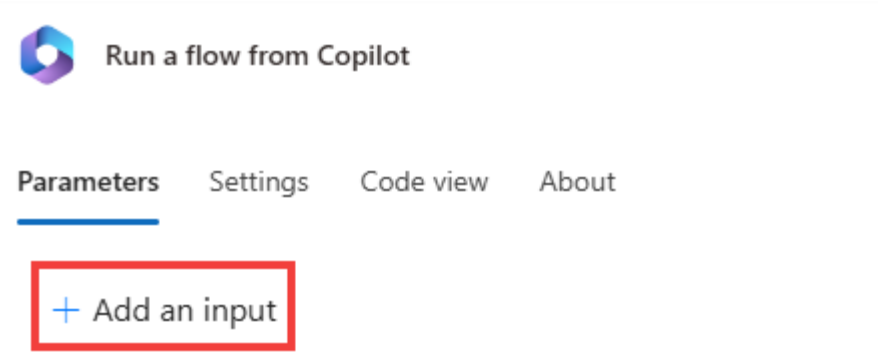


2. This will open **Power Automate** in a new browser window and includes the scaffolding pre and post actions for a new Power Automate cloud flow to interact with Copilot Studio, as shown in the screenshot below:



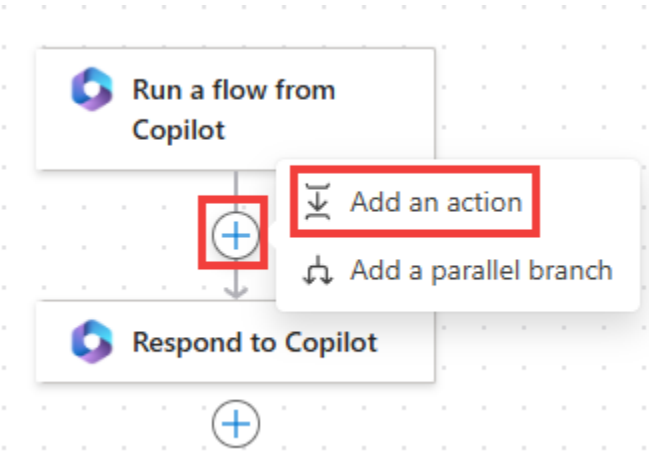
Pro tip: Make sure the **New designer** is enabled in Power Automate, in the top right corner

3. Click on the **Run a flow from Copilot** trigger, in the left-hand pane click **Add an input**



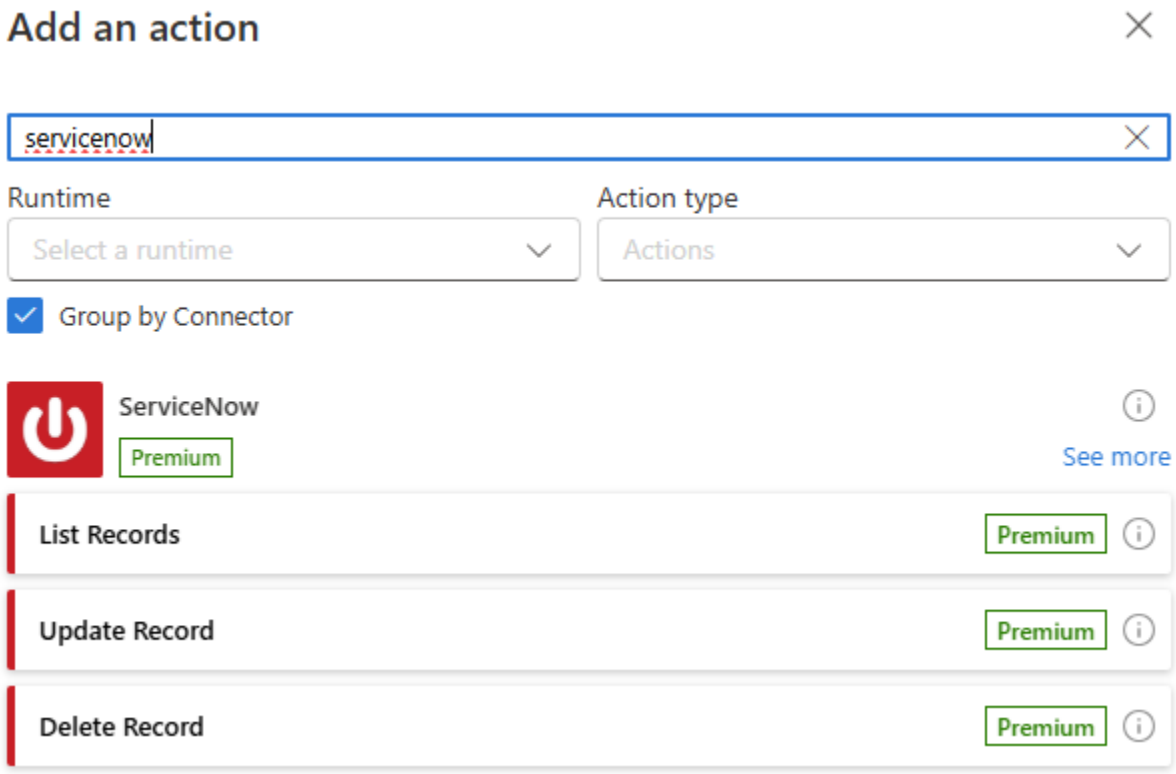
4. Add an **input** of type **Text**, and call it `TicketNumber`

5. Click **Add an action** in between the trigger and the action you just added



Note: in the next steps, **if the ServiceNow environment isn't active** and if the connection cannot be properly established – this is typically the case if the **Record Type** doesn't return a list with **Incident** as an option in step #8 – then **skip to step #11**.

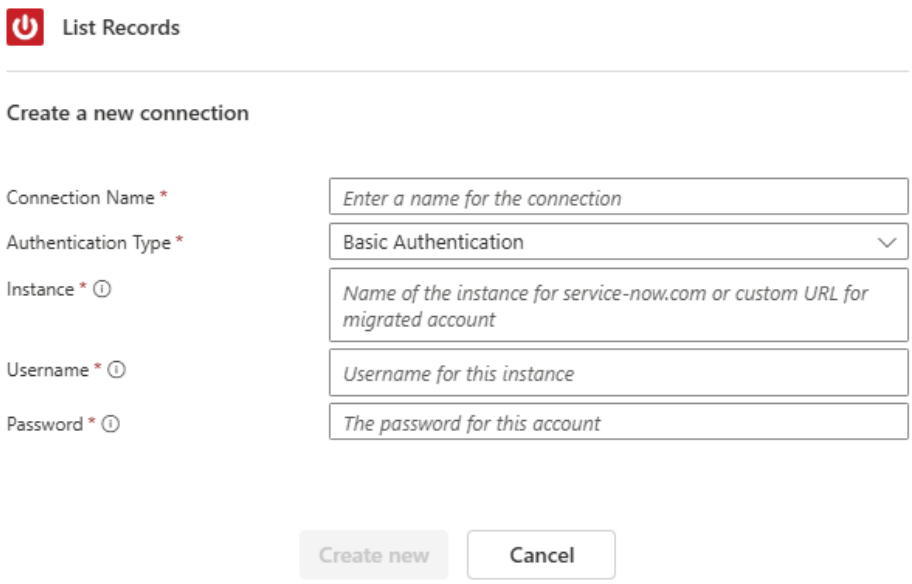
6. Search for `ServiceNow List Records` in the search bar and select **List Records**



Create a connection for **ServiceNow.** Name the connection to something unique, for example `ServiceNow - {Your User Name}`.
Set the **Authentication type** to **Basic Authentication**
Set the **Instance** to `https://dev285430.service-now.com/` Replace your instance ID
Set the **Username** to `CopilotStudioServiceAccount`
Set the **Password** to:

```
}D$wg1TGerz8mg]>L_(!IwamvL,VDo.H>=@!HHfVo+ohu8:%)I4BJBcMEL:&!(S3094egV
:RP}HRQVUp:@1F{!pdJdo5_Ma2ZzJX
```

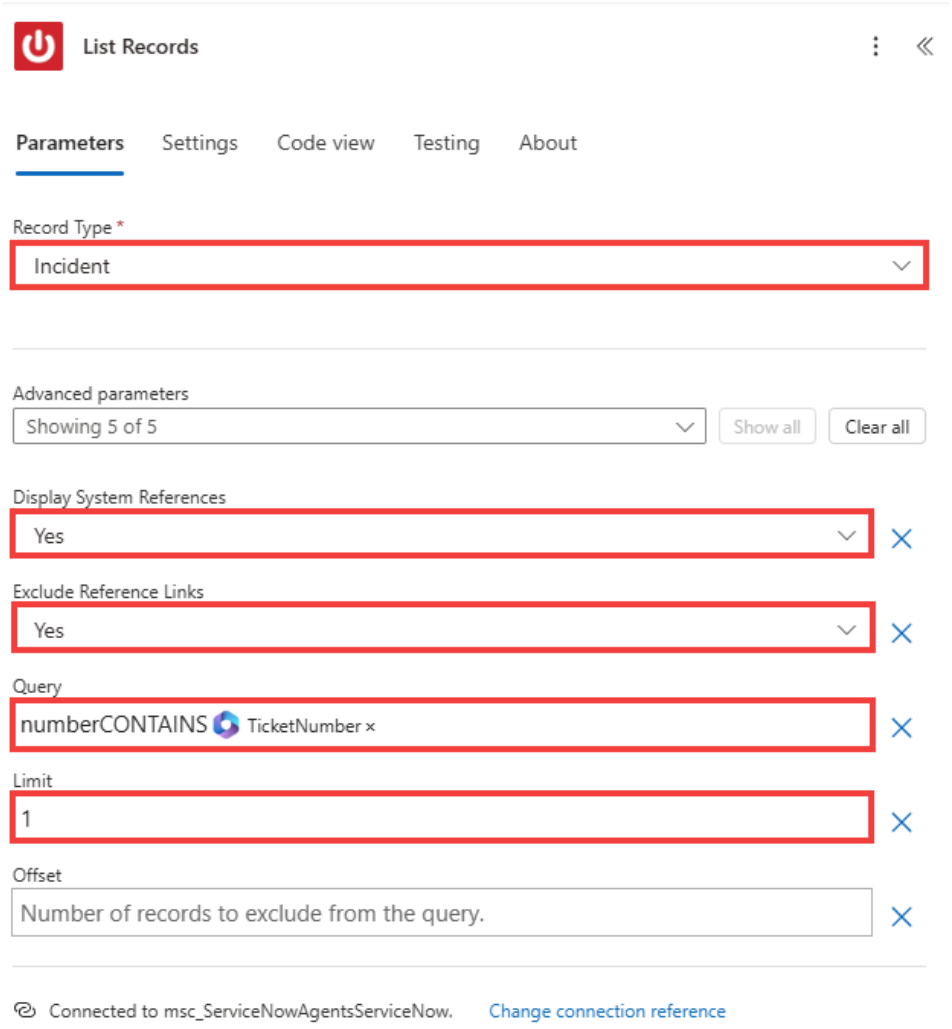
Then select **Create New**



7. Now for **Record Type**, select `Incident` (you can search to make it easier to find Incident in the list).
   Select **Show all** in **Advanced parameters**.
   Set the **Display System References** to `Yes` to show actual values.
   Leave the **Exclude Reference Links** to `Yes`.
   In **Query**, type `numberCONTAINS` and select the `TicketNumber` input from the dynamic content (⚡ icon).
   Ensure there are no spaces between **numberCONTAINS** and the **variable** you reference.
   Alternatively, you can also paste the below in the **Query** field:

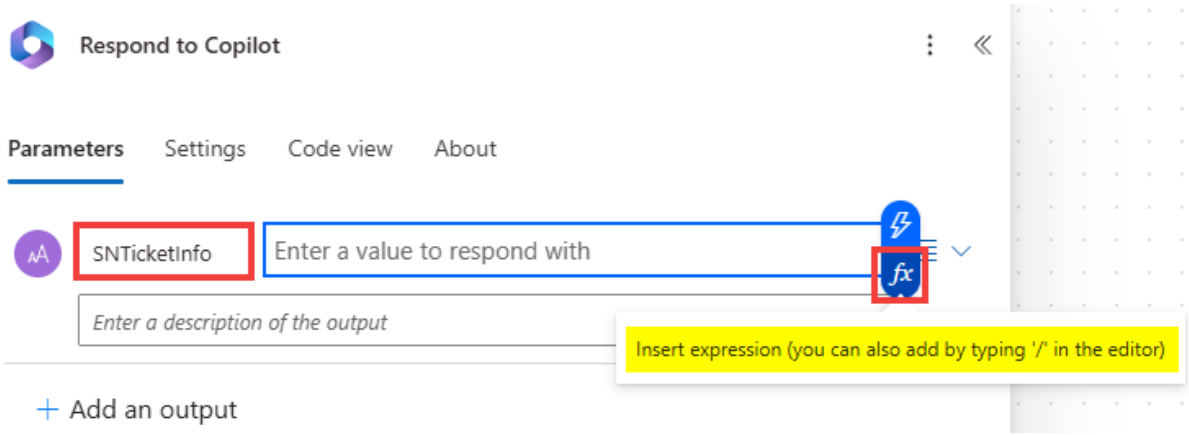
```
numberCONTAINS@{triggerBody()?['text']}
```

   Set Limit to `1`.

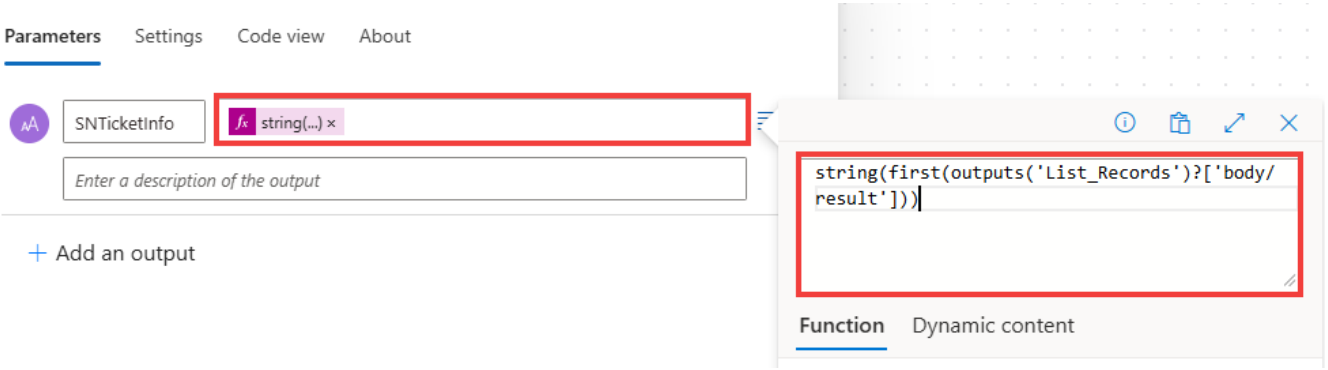


8. Click on the **Respond to Copilot** action in the cloud flow and add an **output** of type **Text** and call it `SNTicketInfo`.

9. For the output **value**, use the formula button fx in order to get a string version of the first returned record of the result array from the List Records body.
Paste the below formula, and select **Add**.




```
string(first(outputs('List_Records')?['body/result']))
```



10. **Optional step (if ServiceNow isn't working):** for the output **value**, paste a the below hardcoded payload. This represents an example of what ServiceNow would typically return.

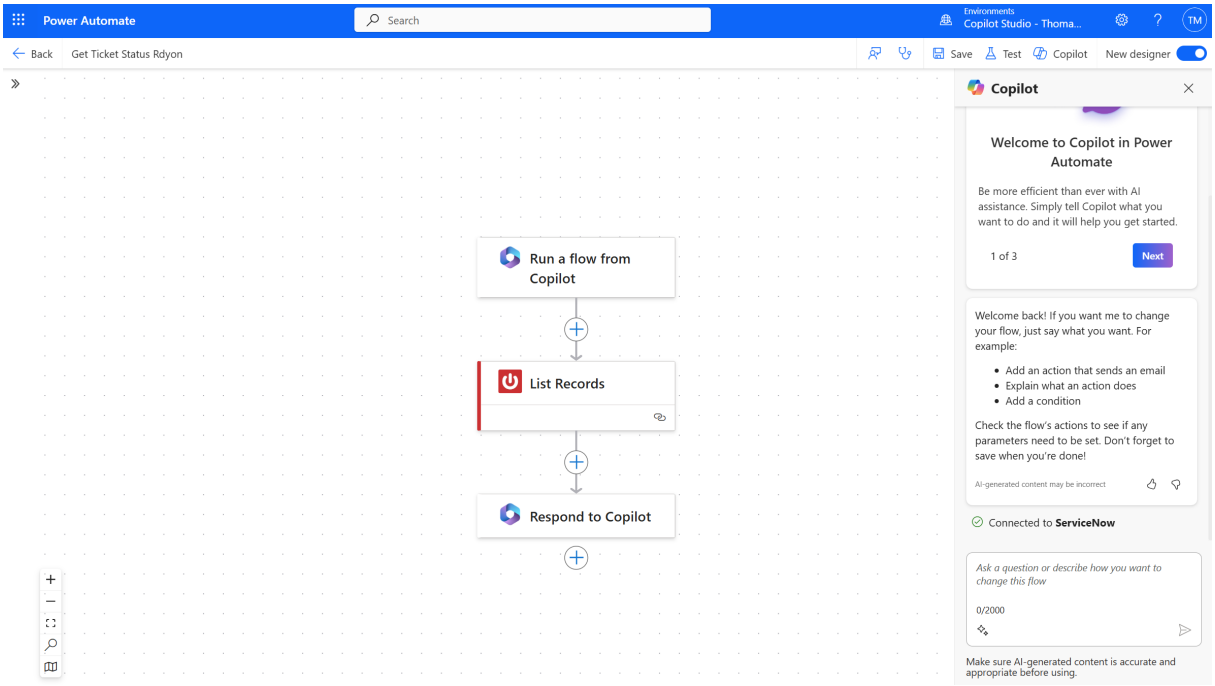
If you struggle copying the below text, go the **Misc** folder in **Lab files** and open ServiceNow Sample JSON Payload.txt

```
{
    "parent": "",
    "made_sla": "true",
    "caused_by": "",
    "watch_list": "",
    "upon_reject": "Cancel all future Tasks",
    "sys_updated_on": "2018-12-12 23:18:55",
    "child_incidents": "0",
    "hold_reason": "",
    "origin_table": "",
    "task_effective_number": "INC0009005",
    "approval_history": "",
    "number": "INC0009005",
    "resolved_by": "",
    "sys_updated_by": "admin",
    "opened_by": "System Administrator",
    "user_input": "",
    "sys_created_on": "2018-08-31 21:35:45",
    "sys_domain": "global",
    "state": "New",
    "route_reason": "",
    "sys_created_by": "admin",
    "knowledge": "false",
    "order": "",
    "calendar_stc": "",
    "closed_at": "",
    "cmdb_ci": "",
    "delivery_plan": "",
    "contract": "",
    "impact": "1 - High",
    "active": "true",
    "work_notes_list": "",
    "business_service": "",
    "business_impact": "",
    "priority": "1 - Critical",
    "sys_domain_path": "/",
    "rfc": "",
    "time_worked": "",
    "expected_start": "",
    "opened_at": "2018-08-31 21:35:21",
    "business_duration": "",
    "group_list": "",
    "work_end": "",
    "caller_id": "David Miller",
    "reopened_time": "",
    "resolved_at": "",
    "approval_set": "",
    "subcategory": "Email",
    "work_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the
criticality of the Incident.\n\n",
    "universal_request": "",
    "short_description": "Email server is down.",
    "correlation_display": "",
    "delivery_task": "",
    "work_start": "",
    "assignment_group": "",
    "additional_assignee_list": "",
    "business_stc": "",
    "cause": "",
    "description": "Unable to send or receive emails.",
    "origin_id": "",
    "calendar_duration": "",
    "close_notes": "",
    "notify": "Do Not Notify",
    "service_offering": "",
    "sys_class_name": "Incident",
    "closed_by": "",
    "follow_up": "",
    "parent_incident": "",
    "sys_id": "ed92e8d173d023002728660c4cf6a7bc",
    "reopened_by": "",
    "incident_state": "New",
    "urgency": "1 - High",
    "problem_id": "",
    "company": "",
    "reassignment_count": "0",
    "activity_due": "2018-12-13 01:18:55",
    "assigned_to": "",
    "severity": "3 - Low",
    "comments": "",
    "approval": "Not Yet Requested",
    "sla_due": "UNKNOWN",
    "comments_and_work_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based
on the criticality of the Incident.\n\n",
    "due_date": "",
    "sys_mod_count": "3",
    "reopen_count": "0",
    "sys_tags": "",
    "escalation": "Normal",
    "upon_approval": "Proceed to Next Task",
    "correlation_id": "",
    "location": "",
    "category": "Software"
}
```


11. The cloud flow is almost complete. It needs to be **renamed** before we move on to best practices, so it is easily found in Copilot Studio and by administrators.
    Click the template **title** as shown in the screenshot below and rename it to `Get Ticket Status {YourUserName}`

12. Click **Publish** and wait a moment until you see the green banner as shown in the screenshot below.

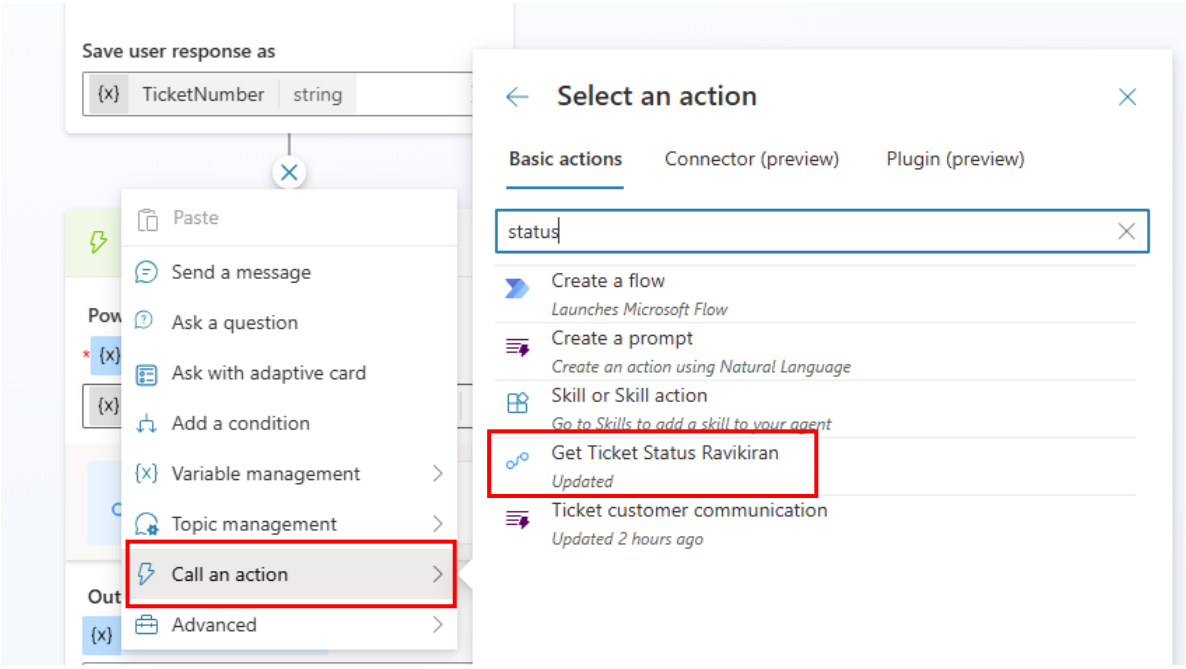
Pro tip: if you don't see the **Publish** button, select **Save**.



We have completed our work in Power Automate. Now let's switch back to Copilot Studio.

Task 3: Call your Power Automate cloud flow from Copilot Studio

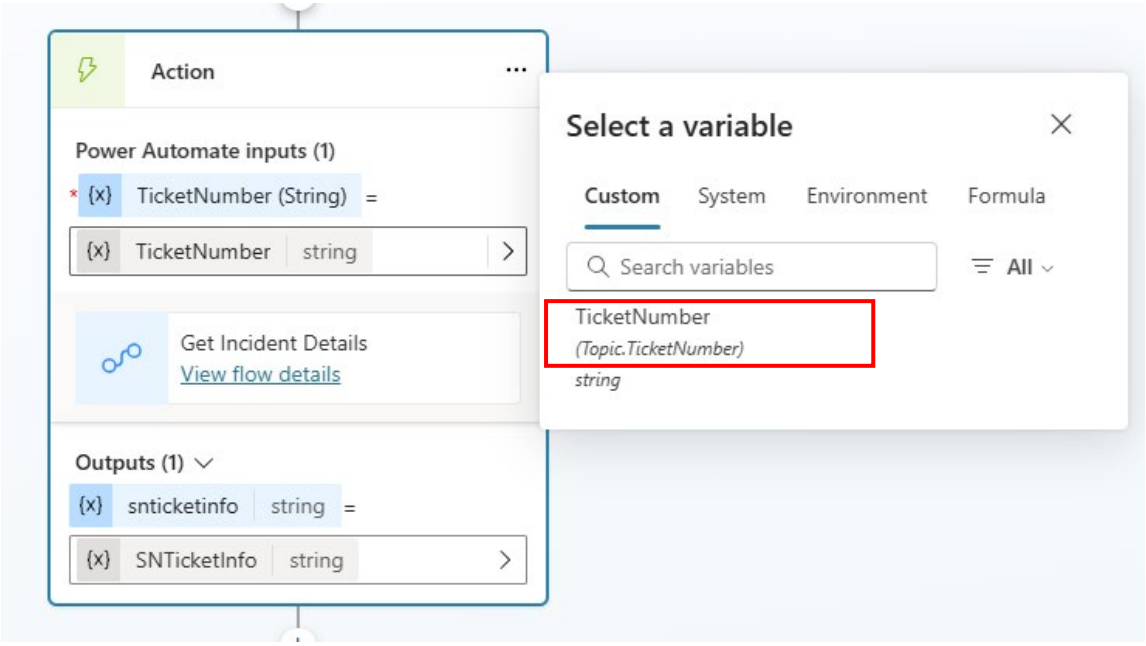
1. Open your existing topic in Copilot Studio and go back to the bottom of your flow, as shown below. Click **Call an action**, and you should see your new Power Automate cloud flow in the list under the **Basic actions** tab. From the list, select your **Get Ticket Status** flow (exact name may differ based on participants).



Pro tip: If you don't see the cloud flow you've created, **Save** the topic, and **refresh** the page to try again.

2. When you select your **Get Ticket Status** flow, you will see a new **Action** node is automatically added.
If the flow requires an input, it requests the value to be selected. As the flow you created in the previous steps requires the **TicketNumber** input, we need to add this input into the Power Automate action by selecting the variable the value is stored in from the user, which is **TicketNumber** from the previous steps earlier in the lab.

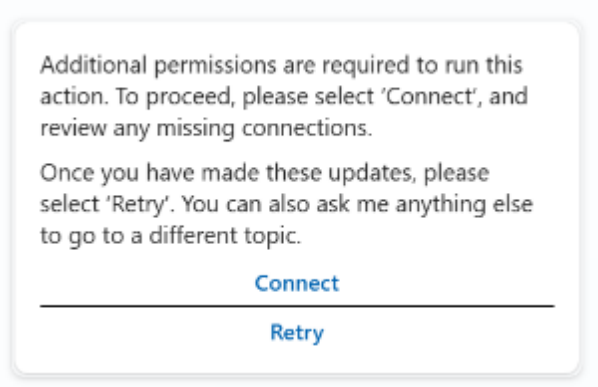
3. Select **Enter or select a value** and choose your **TicketNumber** variable you created in the previous steps of this lab. This is now connected to the Power Automate flow, and outputs the result from Power Automate into the **SNTicketInfo** variable.



Pro tips:

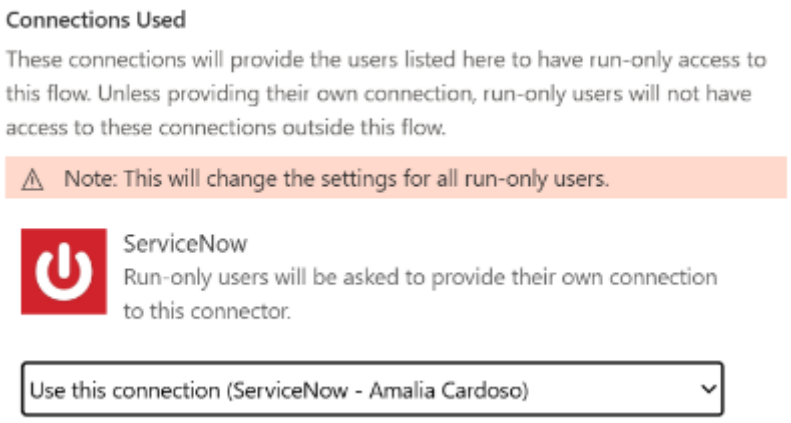
- If a latency is expected from your integration, go the action's properties and add a **Latency Message**, for example: `I'm getting these details for you. Hold on…`
- Consider using **HTTP requests** and **connectors** directly in Microsoft Copilot Studio to avoid the added latency of invoking and running a cloud flow in Power Automate.

4. To use the **author's permissions** (i.e., the connection of the user who created the cloud flow to connect to ServiceNow), and **not the end-user permissions**, follow the below steps. Otherwise, you'll get the below error:



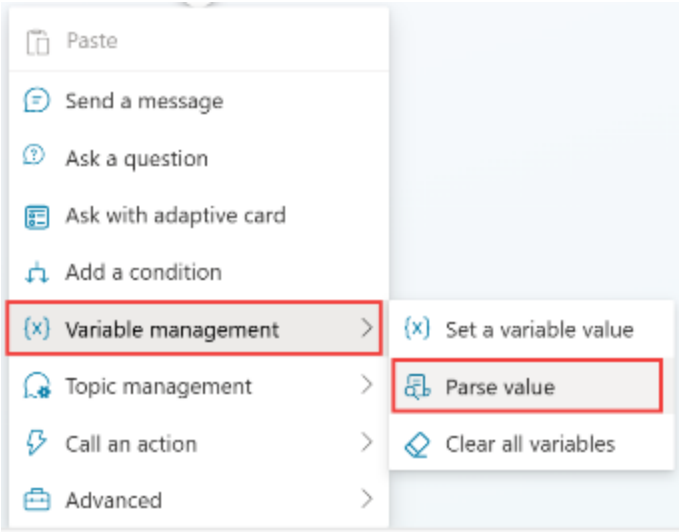
a. In your action step, click on **View flow details**.
b. Edit the **Run only users** option.
c. Switch to "**Use this connection (ServiceNow - {Your User Name})**"

d. Select **Save**.



5. As **ServiceNow** will return the full details of the incident in a technical, **JSON**, format, you need to parse it so that Copilot Studio fully understand its content based on its schema.

6. To do this, **go back to your "Check Ticket Status" topic**, go to the end, **click on (+)**, select **Variable management**, and then **Parse value**.

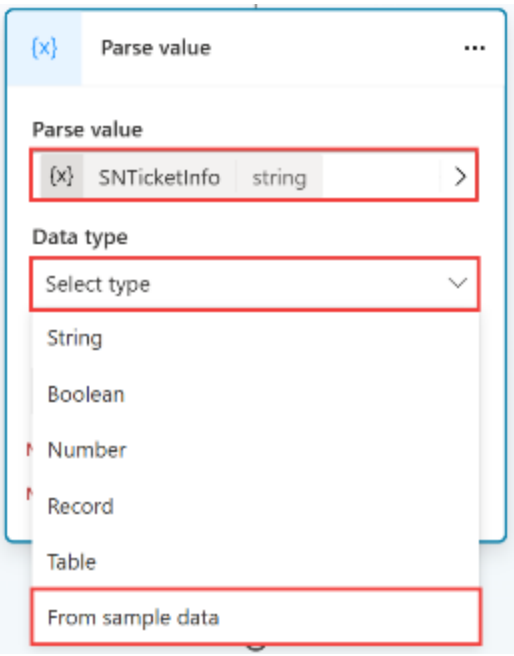


To parse the JSON you can use the Rest API Explorer in ServiceNow to get the structure of the body, or get the schema from a sample payload. For the lab, we're providing sample ServiceNow data later in this chapter.

If you struggle copying the below text, go the **Misc** folder in **Lab files**, and open ServiceNow Sample JSON Payload.txt

7. Now we will need to select the value to parse by selecting the **SNTicketInfo** variable from the Power Automate flow action.

8. Next we will select the **Data type** to **From sample data**



9. Now paste in the JSON sample provided below (next page), after selecting **Get schema from sample JSON** and select **Confirm**.

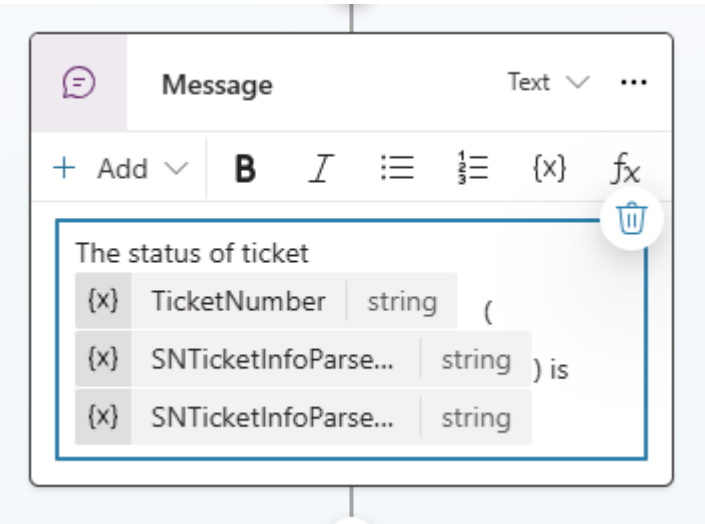
```
{
    "parent": "",
    "made_sla": "true",
    "caused_by": "",
    "watch_list": "",
    "upon_reject": "Cancel all future Tasks",
    "sys_updated_on": "2018-12-12 23:18:55",
    "child_incidents": "0",
    "hold_reason": "",
    "origin_table": "",
    "task_effective_number": "INC0009005",
    "approval_history": "",
    "number": "INC0009005",
    "resolved_by": "",
    "sys_updated_by": "admin",
    "opened_by": "System Administrator",
    "user_input": "",
    "sys_created_on": "2018-08-31 21:35:45",
    "sys_domain": "global",
    "state": "New",
    "route_reason": "",
    "sys_created_by": "admin",
    "knowledge": "false",
    "order": "",
    "calendar_stc": "",
    "closed_at": "",
    "cmdb_ci": "",
    "delivery_plan": "",
    "contract": "",
    "impact": "1 - High",
    "active": "true",
    "work_notes_list": "",
    "business_service": "",
    "business_impact": "",
    "priority": "1 - Critical",
    "sys_domain_path": "/",
    "rfc": "",
    "time_worked": "",
    "expected_start": "",
    "opened_at": "2018-08-31 21:35:21",
    "business_duration": "",
    "group_list": "",
    "work_end": "",
    "caller_id": "David Miller",
    "reopened_time": "",
    "resolved_at": "",
    "approval_set": "",
    "subcategory": "Email",
    "work_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based on the
criticality of the Incident.\n\n",
    "universal_request": "",
    "short_description": "Email server is down.",
    "correlation_display": "",
    "delivery_task": "",
    "work_start": "",
    "assignment_group": "",
    "additional_assignee_list": "",
    "business_stc": "",
    "cause": "",
    "description": "Unable to send or receive emails.",
    "origin_id": "",
    "calendar_duration": "",
    "close_notes": "",
    "notify": "Do Not Notify",
    "service_offering": "",
    "sys_class_name": "Incident",
    "closed_by": "",
    "follow_up": "",
    "parent_incident": "",
    "sys_id": "ed92e8d173d023002728660c4cf6a7bc",
    "reopened_by": "",
    "incident_state": "New",
    "urgency": "1 - High",
    "problem_id": "",
    "company": "",
    "reassignment_count": "0",
    "activity_due": "2018-12-13 01:18:55",
    "assigned_to": "",
    "severity": "3 - Low",
    "comments": "",
    "approval": "Not Yet Requested",
    "sla_due": "UNKNOWN",
    "comments_and_work_notes": "2018-12-12 23:18:42 - System Administrator (Work notes)\nupdated the priority to high based
on the criticality of the Incident.\n\n",
    "due_date": "",
    "sys_mod_count": "3",
    "reopen_count": "0",
    "sys_tags": "",
    "escalation": "Normal",
    "upon_approval": "Proceed to Next Task",
    "correlation_id": "",
    "location": "",
    "category": "Software"
}
```


10. Then set a variable to store the parsed record for later user.
    Create a variable called `SNTicketInfoParsed`. Its type will automatically be set based on its schema (record).

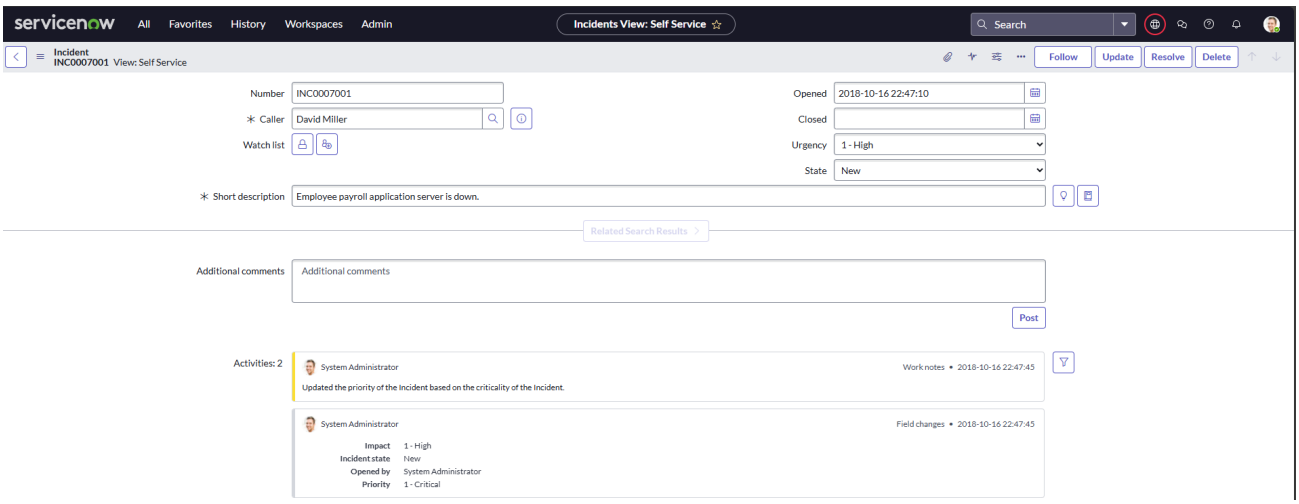
11. Now, add a response to give the user of your agent a formatted response to tell them the status of the ticket. Click the **(+)** to add a new node and select **Send a message**.
Bold the key information either with the command bar or by surrounding the text with **.

```
The status of ticket {Topic.TicketNumber}
({Topic.SNTicketInfoParsed.short_description}) is
{Topic.SNTicketInfoParsed.state}
```

Pro tip: Copilot Studio and some channels support [Markdown] for simple formatting.



You can look above at the sample JSON to see what data would be returned in what value, however below is a screenshot of the ticket we will lookup later to help you with structuring your message:

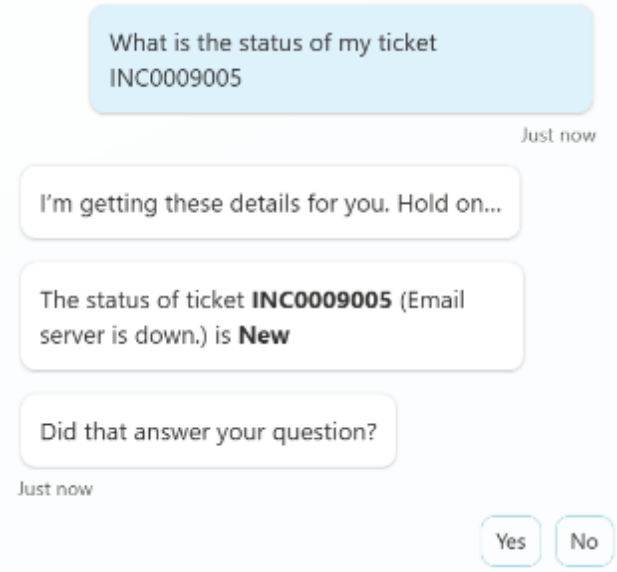


12. To end the conversation, select **(+) to add a new node**, **Topic management** and **Go to another topic** and select **End of Conversation**.

13. **Save** your topic and **test** your copilot.

```
What is the status of my ticket INC0007001?
```

14. Make sure you check the status of ticket number **INC0007001**:

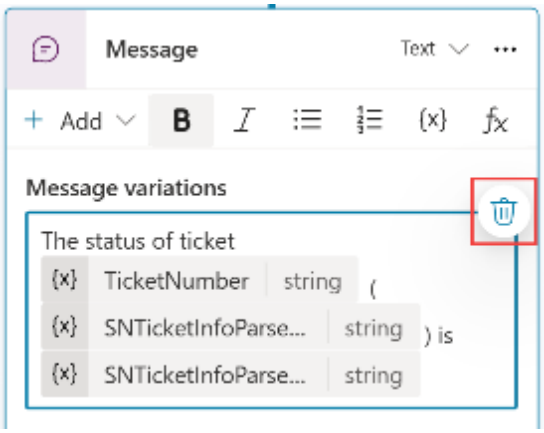


You have successfully created a Power Automate cloud flow and a new topic in Microsoft Copilot Studio that used the flow to provide real time data from an external service to the user!

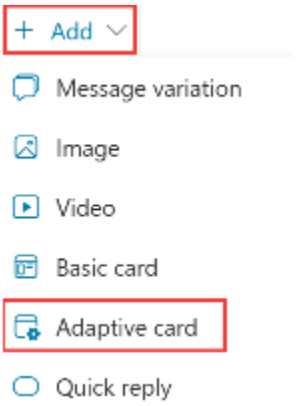
Task 4: Display the ServiceNow ticket information in an adaptive card

1. Go to your **Check Ticket Status** topic

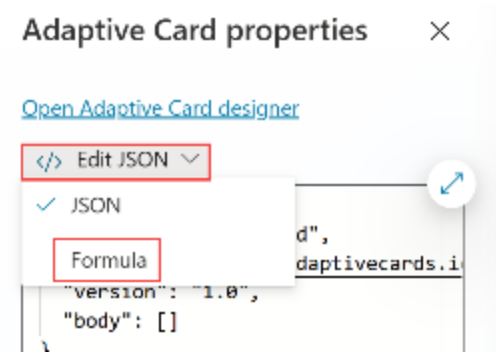
2. **Delete** the text message variable for your message



3. Select **Add**, **Adaptive Card**



4. Toggle from **Edit JSON** to **Formula**, so that you can make the adaptive card dynamic and author it in Power Fx language.



5. **Paste** the below Power Fx formula that already contains the references to the ServiceNow ticket information

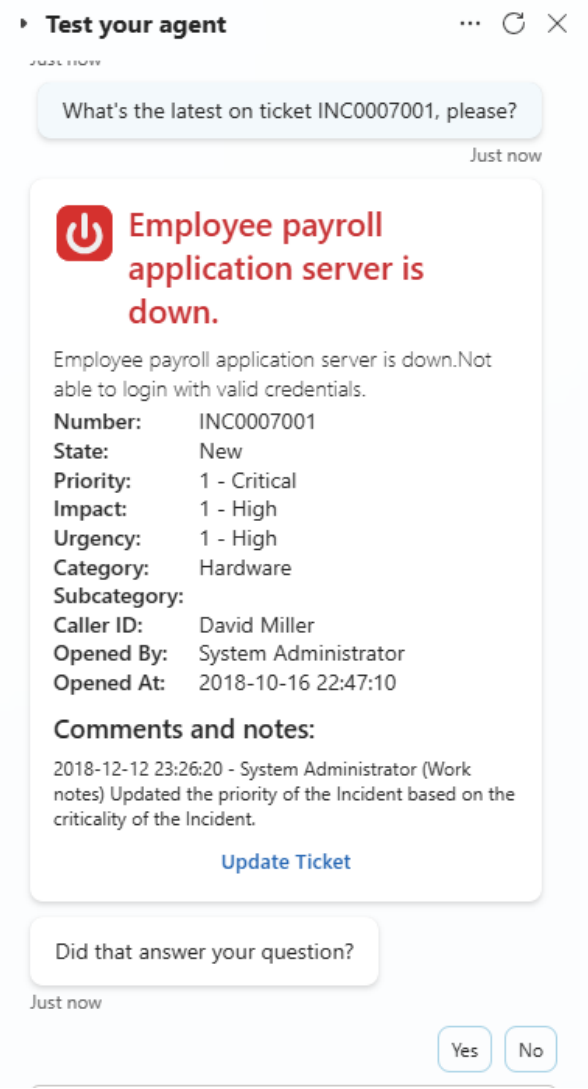
If you struggle copying the below text, go the **Misc** folder in **Lab files**, and open ServiceNow Adaptive Card Power Fx.txt

```
{
  type: "AdaptiveCard",
  version: "1.5",
  body: [
    {
      type: "ColumnSet",
      columns: [
        {
          type: "Column",
          width: "auto",
          items: [
            {
              type: "Image",
              url: "https://upload.wikimedia.org/wikipedia/commons/6/67/Shutdown_button_red_wikimedia.svg",
              size: "Small"
            }
          ]
        },
        {
          type: "Column",
          width: "stretch",
          items: [
            {
              type: "TextBlock",
              text: Topic.SNTicketInfoParsed.short_description,
              weight: "Bolder",
              size: "Large",
              wrap: true,
              color: "Attention",
              horizontalAlignment: "Left"
            }
          ],
          verticalContentAlignment: "Center",
          horizontalAlignment: "Center"
        }
      ]
    },
    {
      type: "TextBlock",
      text: Topic.SNTicketInfoParsed.description,
      weight: "Lighter",
      wrap: true
    },
    {
      type: "FactSet",
      facts: [
        {
          title: "Number:",
          value: Topic.SNTicketInfoParsed.number
        },
        {
          title: "State:",
          value: Topic.SNTicketInfoParsed.state
        },
        {
          title: "Priority:",
          value: Topic.SNTicketInfoParsed.priority
        },
        {
          title: "Impact:",
          value: Topic.SNTicketInfoParsed.impact
        },
        {
          title: "Urgency:",
          value: Topic.SNTicketInfoParsed.urgency
        },
        {
          title: "Category:",
          value: Topic.SNTicketInfoParsed.category
        },
        {
          title: "Subcategory:",
          value: Topic.SNTicketInfoParsed.subcategory
        },
        {
          title: "Caller ID:",
          value: Topic.SNTicketInfoParsed.caller_id
        },
        {
          title: "Opened By:",
          value: Topic.SNTicketInfoParsed.opened_by
        },
        {
          title: "Opened At:",
          value: Topic.SNTicketInfoParsed.opened_at
        }
      ],
      spacing: "Small"
    },
    {
      type: "TextBlock",
      text: "Comments and notes:",
      weight: "Bolder",
      size: "Medium",
      wrap: true
    },
    {
      type: "TextBlock",
      text: Topic.SNTicketInfoParsed.comments_and_work_notes,
      wrap: true,
      size: "Small"
    }
  ],
  actions: [
    {
      type: "Action.OpenUrl",
      title: "Update Ticket",
      url: "https://dev204932.service-now.com/nav_to.do?uri=incident.do?sys_id=" & Topic.SNTicketInfoParsed.sys_id & "%26sysparm_view=ess"
    }
  ],
  '$schema': "http://adaptivecards.io/schemas/adaptive-card.json"
}
```

6. **Save**.

7. **Test** your agent.

```
What's the latest on ticket INC0007001, please?
```



If the link to update your ticket does not work you might need to replace the ServiceNow URL in the adaptive card by the latest one used in your workshop.

Summary

Thank you for completing the lab 'Build and calling Power Automate cloud flows from your copilot'. You have successfully:

- Created a new Power Automate cloud flow
- Called the Power Automate cloud flow into your topic
- Set input and output variables
- Displayed dynamic data back to the user in Copilot Studio

Terms of Use

By using this document, in whole or in part, you agree to the following terms:

Notice

Information and views expressed in this document, including (without limitation) URL and other Internet Web site references, may change without notice. Examples depicted herein, if any, are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product.

Use Limitations

Copying or reproduction, in whole or in part, of this document to any other server or location for further reproduction or redistribution is expressly prohibited. Microsoft provides you with this document for purposes of obtaining your suggestions, comments, input, ideas, or know-how, in any form, ("Feedback") and to provide you with a learning experience. You may use this document only to evaluate its content and provide feedback to Microsoft. You may not use this document for any other purpose. You may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this document or any portion thereof. You may copy and use this document for your internal, reference purposes only.

Feedback

If you give Microsoft any Feedback about this document or the subject matter herein (including, without limitation, any technology, features, functionality, and/or concepts), you give to Microsoft, without charge, the right to use, share, and freely commercialize Feedback in any way and for any purpose. You also give third parties, without charge, the right to use, or interface with, any Microsoft products or services that include the Feedback. You represent and warrant that you own or otherwise control all rights to such Feedback and that no such Feedback is subject to any third-party rights.

DISCLAIMERS

CERTAIN SOFTWARE, TECHNOLOGY, PRODUCTS, FEATURES, AND FUNCTIONALITY (COLLECTIVELY "CONCEPTS"), INCLUDING POTENTIAL NEW CONCEPTS, REFERENCED IN THIS DOCUMENT ARE IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION AND ARE INTENDED FOR FEEDBACK AND TRAINING PURPOSES ONLY. THE CONCEPTS REPRESENTED IN THIS DOCUMENT MAY NOT REPRESENT FULL FEATURE CONCEPTS AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. MICROSOFT ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH CONCEPTS. YOUR EXPERIENCE WITH USING SUCH CONCEPTS IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

THIS DOCUMENT, AND THE CONCEPTS AND TRAINING PROVIDED HEREIN, IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS, IMPLIED, OR STATUTORY, INCLUDING (WITHOUT LIMITATION) THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NONINFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, THE OUTPUT THAT DERIVES FROM USE OF THIS DOCUMENT OR THE CONCEPTS, OR THE SUITABILITY OF THE CONCEPTS OR INFORMATION CONTAINED IN THIS DOCUMENT FOR ANY PURPOSE.

MICROSOFT COPILOT STUDIO (1) IS NOT INTENDED OR MADE AVAILABLE AS A MEDICAL DEVICE FOR THE DIAGNOSIS OF DISEASE OR OTHER CONDITIONS, OR IN THE CURE, MITIGATION, TREATMENT OR PREVENTION OF DISEASE, OR OTHERWISE TO BE USED AS A COMPONENT OF ANY CLINICAL OFFERING OR PRODUCT, AND NO LICENSE OR RIGHT IS GRANTED TO USE MICROSOFT COPILOT STUDIO FOR SUCH PURPOSES, (2) IS NOT DESIGNED OR

INTENDED TO BE A SUBSTITUTE FOR PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, TREATMENT, OR JUDGMENT AND SHOULD NOT BE USED AS A SUBSTITUTE FOR, OR TO REPLACE, PROFESSIONAL MEDICAL ADVICE, DIAGNOSIS, TREATMENT, OR JUDGMENT, AND (3) SHOULD NOT BE USED FOR EMERGENCIES AND DOES NOT SUPPORT EMERGENCY CALLS. ANY CHATBOT YOU CREATE USING MICROSOFT COPILOT STUDIO IS YOUR OWN PRODUCT OR SERVICE, SEPARATE AND APART FROM MICROSOFT COPILOT STUDIO. YOU ARE SOLELY RESPONSIBLE FOR THE DESIGN, DEVELOPMENT, AND IMPLEMENTATION OF YOUR CHATBOT (INCLUDING INCORPORATION OF IT INTO ANY PRODUCT OR SERVICE INTENDED FOR MEDICAL OR CLINICAL USE) AND FOR EXPLICITLY PROVIDING END USERS WITH APPROPRIATE WARNINGS AND DISCLAIMERS PERTAINING TO USE OF YOUR CHATBOT. YOU ARE SOLELY RESPONSIBLE FOR ANY PERSONAL INJURY OR DEATH THAT MAY OCCUR AS A RESULT OF YOUR CHATBOT OR YOUR USE OF MICROSOFT COPILOT STUDIO IN CONNECTION WITH YOUR CHATBOT, INCLUDING (WITHOUT LIMITATION) ANY SUCH INJURIES TO END USERS.