

LAB-1309

Decoding AI in Webex Contact Center: Best Practices, Pitfalls, and Pro Tips

Anuj Bhatia, D.K. Singh

Copyright © 2025 Cisco

Table of contents

1. Lab	3
1.1 Overview	3
1.2 Excercise 1 - Build Autonomous AI Agents with Knowledge Bases	4
1.3 Result	21
1.4 Task 2 - Build Autonomous Webex AI Agent Actions	22
1.5 Lab 3 - Create and Test Voice Flows for Webex AI Agents	40
1.6 Excercise 4 - Explore Webex AI Assistant Feature	49
1.7 Results	61
1.8 Excercise 5 - (Bonus) Car Order Actions with Webex AI Agent	62
1.9 Conclusion	69
2. How to use this guide	70
2.1 Setting up and using the tool	70
2.2 Styling and Advanced Features	74

1. Lab

1.1 Overview

1.1.1 Learning Objectives

This lab series focuses on building, configuring, and testing Autonomous Webex AI Agents. You'll start by setting up these agents and their knowledge base, enabling them to answer queries from a knowledge repository.

The labs then guide you through creating and managing agent actions using Webex Connect flows, which allow the AI to interact with external APIs and fulfill requests. You'll also learn to configure a voice flow and test the AI agent's functionality by making a phone call.

Finally, the series highlights the AI Assistant feature within the Webex Contact Center. You'll explore how this AI enhances the live agent experience by providing conversation summaries when a call is transferred or disconnected and offering real-time transcriptions to improve clarity and efficiency.

1.1.2 Disclaimer

Although the lab design and configuration examples could be used as a reference, for design related questions please contact your representative at Cisco, or a Cisco partner.

1.1.3 Getting Started

This lab leverages Webex Contact Center cloud environment.

1.2 Excercise 1 - Build Autonomous AI Agents with Knowledge Bases

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

Control Hub URL	https://admin.webex.com
Username	labuserID@wx1.wbx.ai (where ID is your assigned pod number; this ID will be provided by your proctor)
Password	webexONE1!

AI Agents Overview

Cisco Webex AI Agents provide automated assistance to customers before they interact with human agents across voice or digital channels. These agents leverage language understanding and contextual awareness within conversations. Residing within the CPaaS ecosystem, administrators can utilize the Webex AI Agent Studio platform to design, create, manage, and deploy these agents to meet customer service and support needs.

Two primary types of AI Agents are currently available:

- **Autonomous AI Agents:** These operate independently, making decisions and performing tasks without direct human intervention where the key capabilities include:
 - Answering Questions: Accessing and utilizing a knowledge repository to provide informative and accurate responses to user queries.
 - Performing Actions: Automating time-consuming or repetitive tasks by making choices based on information and predefined rules.
- **Scripted AI Agents:** These deliver precise and timely responses to customer inquiries in real-time. They execute tasks based on specific customer requests or inputs.

Integrating these agents into customer business ecosystems can significantly enhance customer experience by delivering personalized, real-time conversational interactions. Furthermore, AI Agents can be easily scaled to accommodate high volumes of customer interactions without the need for additional human agents, leading to improved customer satisfaction and reduced operational costs.

Prerequisites

To ensure the WxCC Tenant can be enabled for this feature and that the feature can be used effectively, the following requirements must be adhered to:

- Webex Contact Center tenant with **Flex 3** subscription and **A-FLEX-AI-AGT SKU**
- Webex Connect capability on the tenant.

1.2.1 Objective

In this lab, participants will learn to:

- Set up Autonomous AI Agents from scratch.
- Configure these agents to generate and present answers to queries by accessing and utilizing a knowledge repository.
- Create and manage the underlying knowledge base

1.2.2 Section 1 : Create Webex AI Agent and Knowledge Base

- Since the task is to create autonomous AI agents that are capable of answering questions from a knowledge base, the first step would be to build that knowledge base.
- Sign in to Control Hub using the URL <https://admin.webex.com>.

- Use the credentials highlighted in the table above.
- In Quick Links on the right pane of the Contact Center suite section, click Webex AI Agent to access the studio.

The screenshot shows the webex Control Hub interface. On the left, there's a navigation sidebar with sections like Main Menu, Contact Center (Overview selected), Customer Experience (Channels, Queues, Business Hours, Audio Prompts, Flows, Call Recording Schedules, Surveys), and Digital Settings. The main content area is titled 'Contact Center Overview' and includes sections for 'What's new' (Multimedia Profiles, Sites, Teams, Skill Profiles, Desktop Profiles) and a 'View daily details' button. On the right, a 'Quick Links' sidebar lists items under 'Contact Center Suite' (Desktop, Analyzer, Create new flow, Webex Contact Center Management Portal, Topic Analytics, Webex AI Agent - highlighted with a red box), 'Digital Channels' (Webex Connect, Webex Engage), and other icons.

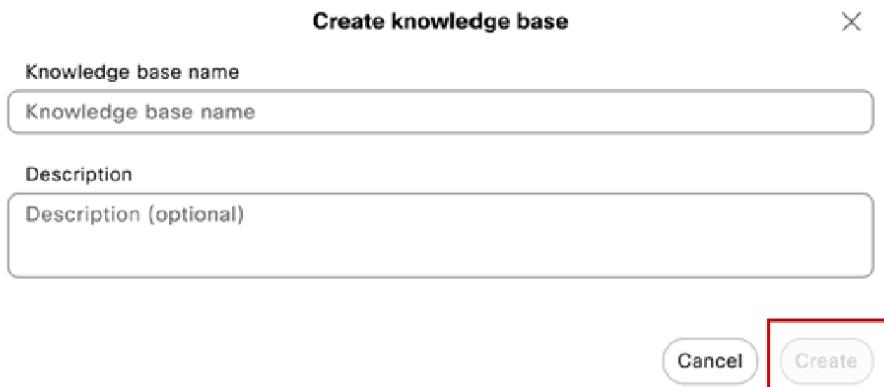
- On the Dashboard, click the Knowledge icon on the left navigation pane to go to the Knowledge Bases page.

The screenshot shows the 'Knowledge bases' page. The left sidebar has icons for Knowledge (highlighted with a red box), Dashboards, and Help. The main area has a search bar ('Search for knowledge base') and a 'KB type' dropdown. Below is a card for a knowledge base named 'CoService' (Updated on 18 Jan, 25), showing 1 document and 0 files. The 'Knowledge' icon in the sidebar is also highlighted with a red box.

- Click + Create Knowledge Base on the upper right corner.



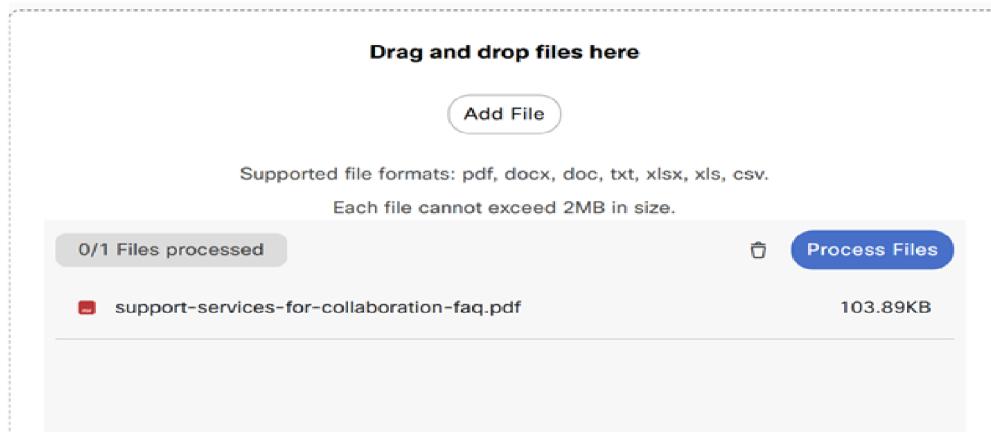
- On the Create Knowledge Base page, enter the knowledge base name as **WebexOne_KB_YourName**.
- Click Create. The system creates a knowledge base with the specified name.



- For this lab, let's add FAQ content specific to buying a cae. Download the word doc from the link below.

Buying Car FAQ

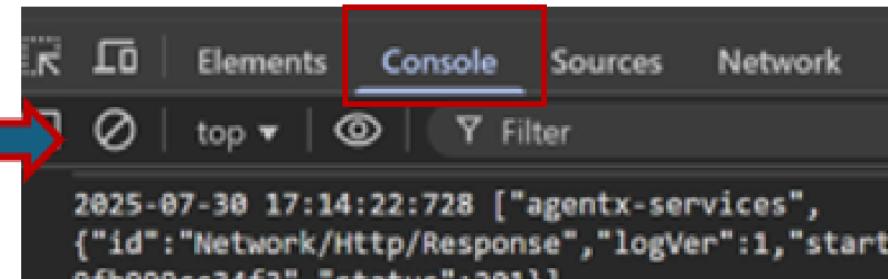
- Drag and drop the file to the knowledge base. Alternatively, click Add File to add a file.
- **DO Not Click on Process Button Yet**



 **Note**

The system supports following file formats: PDF, DOC, DOCX, DOCX, TXT, XLSX, XLS and CSV.
TXT file size cannot exceed 2 MB. All other file size cannot exceed 10MB.

- After adding the file, bring up the browser developer tool (Shortcut: Press F12 Key).
- Once the developer tool is up, ensure that it's on the "Console" tab.
- Clear the console logs by selecting the "Clear Console" button and click Process File to process them.



Drag and drop files here

Add File

Supported file formats: pdf, docx, doc, txt, xlsx, xls, csv.
Each file cannot exceed 2MB in size.

0/1 Files processed

support-services-for-collaboration-faq.pdf 103.89KB

Process Files

- When done, the system lists the files under the Files tab.

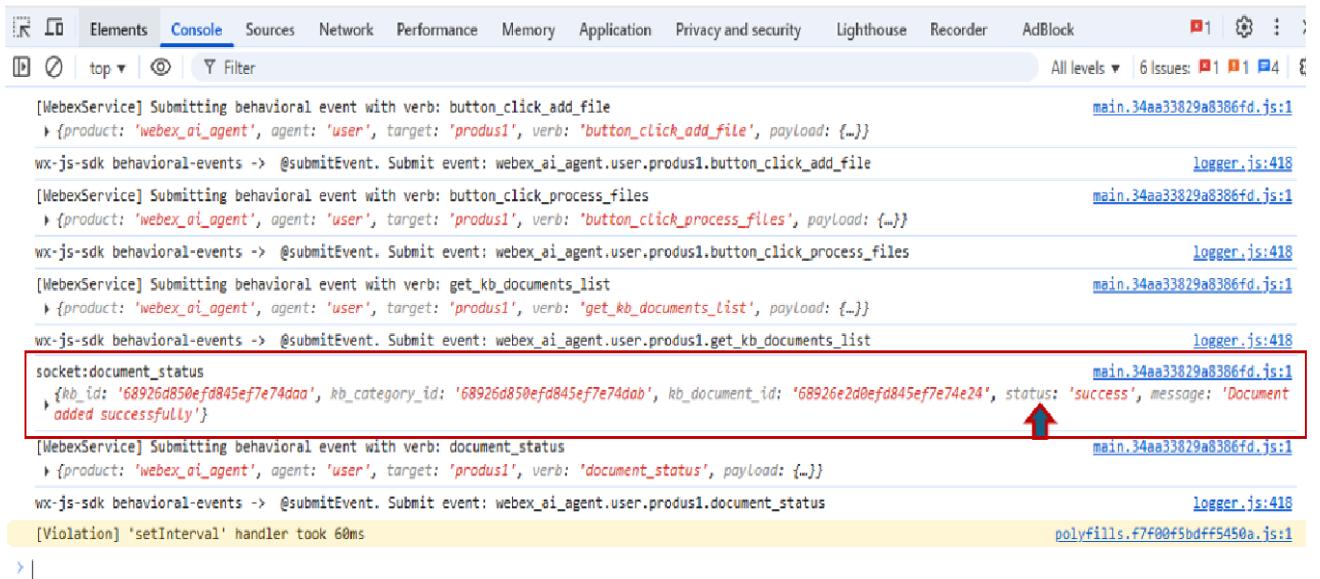
CiscoLive<anuj>

Files Documents Information

Processed files

support-services-for-collaboration-faq.pdf 103.89KB, Uploaded on Jan 19, 2025, 10:57:52 AM

- The browser logs will show a 'success' status, along with the 'kb_id', 'kb_category_id', and 'kb_documented_id'.



```
[WebexService] Submitting behavioral event with verb: button_click_add_file main.34aa33829a8386fd.js:1
  ↪ {product: 'webex_ai_agent', agent: 'user', target: 'produs1', verb: 'button_click_add_file', payload: {}}
wx-js-sdk behavioral-events -> @submitEvent. Submit event: webex_ai_agent.user.produs1.button_click_add_file logger.js:418
[WebexService] Submitting behavioral event with verb: button_click_process_files main.34aa33829a8386fd.js:1
  ↪ {product: 'webex_ai_agent', agent: 'user', target: 'produs1', verb: 'button_click_process_files', payload: {}}
wx-js-sdk behavioral-events -> @submitEvent. Submit event: webex_ai_agent.user.produs1.button_click_process_files logger.js:418
[WebexService] Submitting behavioral event with verb: get_kb_documents_list main.34aa33829a8386fd.js:1
  ↪ {product: 'webex_ai_agent', agent: 'user', target: 'produs1', verb: 'get_kb_documents_list', payload: {}}
wx-js-sdk behavioral-events -> @submitEvent. Submit event: webex_ai_agent.user.produs1.get_kb_documents_list logger.js:418
socket:document_status
  ↪ {kb_id: '68926d850ef845ef7e74daa', kb_category_id: '68926d850ef845ef7e74dab', kb_document_id: '68926e2d0ef845ef7e74e24', status: 'success', message: 'Document added successfully'}
```

 **Note**

This is essential information for Cisco TAC team to analyze and troubleshoot any issues that may have caused the upload to fail.

- Navigate to the Information tab to view and track the details of the files you uploaded and the documents you created , **KB ID** is also present on the top left corner.

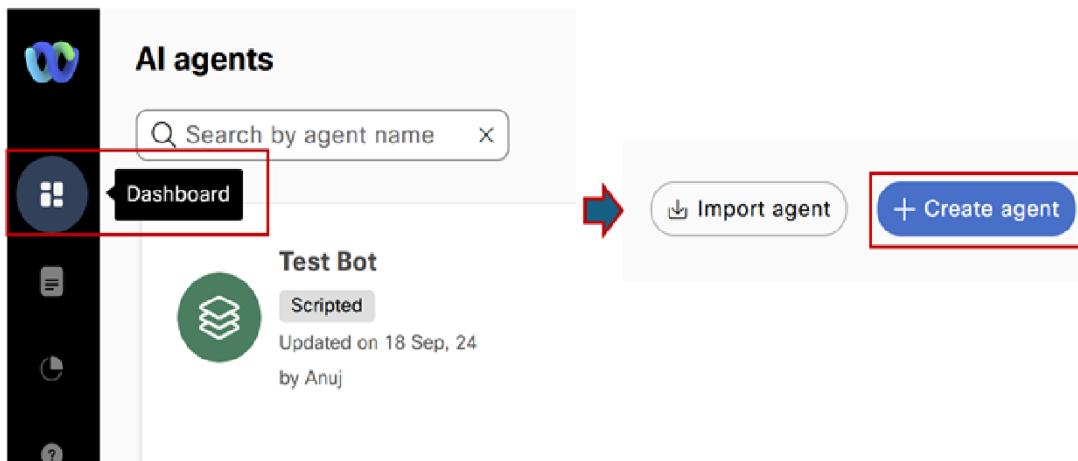
The screenshot shows the 'Information' tab selected in the navigation bar. Key details displayed include:

- KB ID:** 678d16ca8a47fb6887afff63
- Storage:** 103.89KB / 2GB
- Files:** 1 / 20
- Created at:** Jan 19, 2025, 10:14:18 AM
- Created by:** Anuj
- Last updated at:** Jan 19, 2025, 10:57:52 AM
- Last updated by:** Anuj

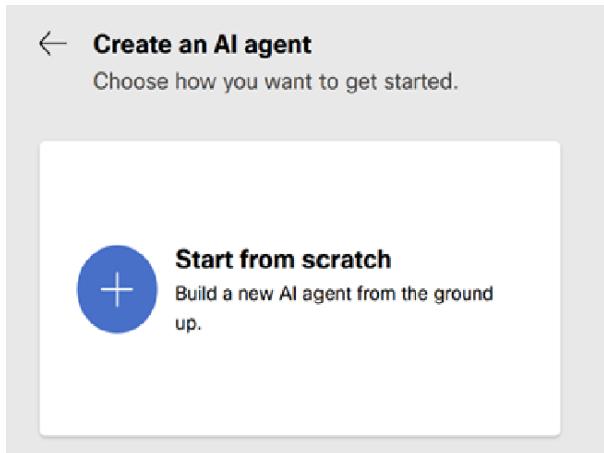
Activity history:

Action	Details	Created/Updated at	Created/Updated by
Created	support-services-for-collaboration-faq.pdf uploaded	19 Jan 25, 10:57 AM	Anuj
Created	KB created	19 Jan 25, 10:14 AM	Anuj

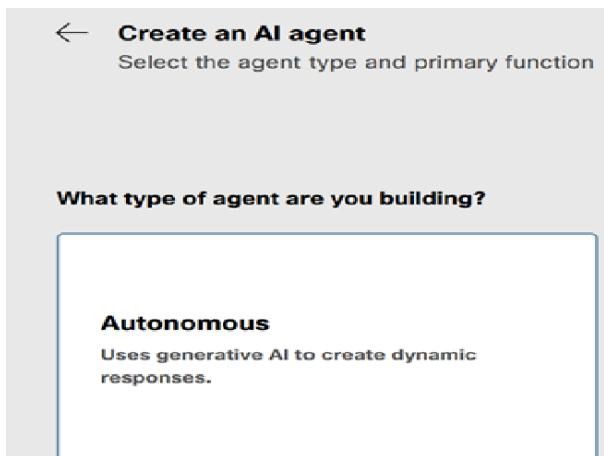
- Now, it's time to create the AI agent itself.
- On the Dashboard, click +Create agent.



- On the Create an AI Agent screen, click **Start from scratch** tile followed by Next button (bottom of the screen)



- In the What type of agent are you building section, click Autonomous.



- On the 'Add the essential details' specify the following details:
- **Agent name:** Enter the name of the AI agent such as **WebexOne_yourname**
- **System ID:** A system-generated unique identifier. This field is editable.
- **AI engine:** Select the required AI engine from the drop-down list (default its Webex AI Pro 1.0).
- **Agent's goal:** Provide a brief description of the AI agent's goal which in this case is to answer queries hence the prompt "**Answer queries about cars**".
- Click Create button at the bottom of the screen.

Add the essential details

Agent name * X

Wx1-DKauto007

System ID *

Wx1-DKauto007-ucZKCpCh X

AI engine * ▼

Webex AI Pro 1.0

Ideal for most contact center use cases with global language support and human-like interactions.

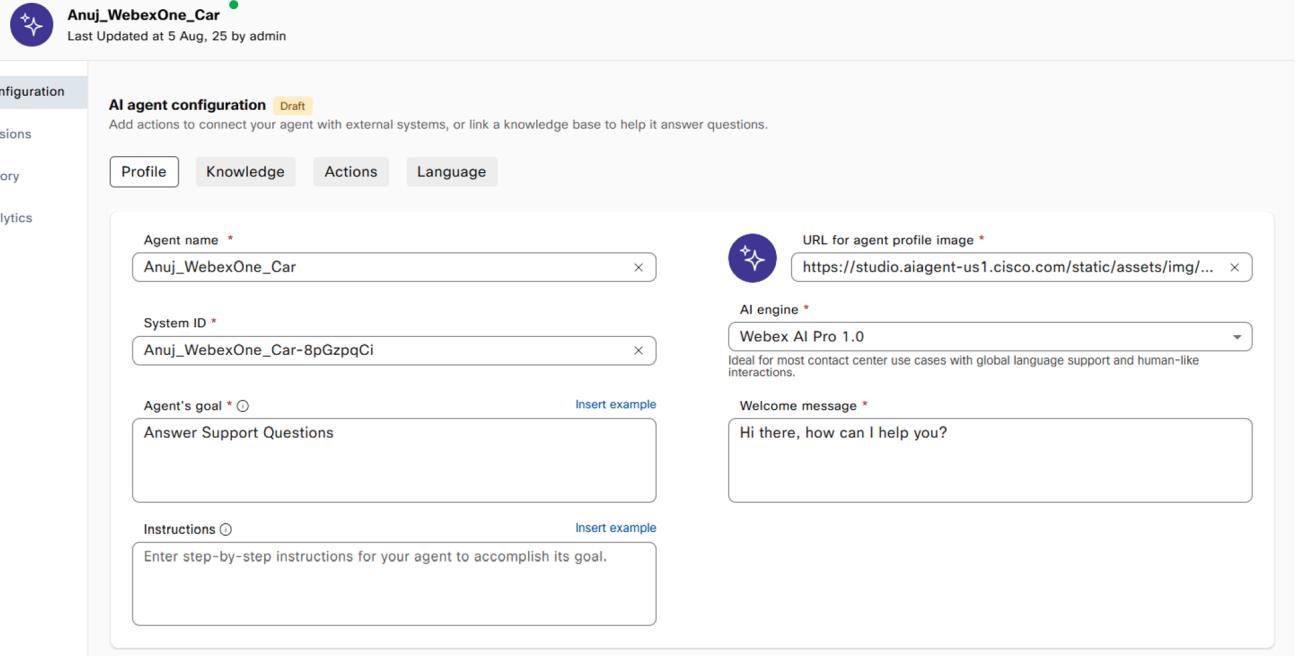
Agent's goal * ⓘ Insert example

Provide answers for ques related to Cisco Collab support.

 **Note**

Create button is grayed out until all fields are populated.

- The autonomous AI agent for answering questions is successfully created and is now available on the Dashboard.



The screenshot shows the 'AI agent configuration' page for an AI agent named 'Anuj_WebexOne_Car'. The page includes tabs for Profile, Knowledge, Actions, and Language. The Profile tab is active. It displays the following fields:

- Agent name ***: Anuj_WebexOne_Car
- System ID ***: Anuj_WebexOne_Car-8pGzpqCi
- Agent's goal * ⓘ**: Answer Support Questions
- Instructions ⓘ**: Enter step-by-step instructions for your agent to accomplish its goal.
- URL for agent profile image ***: https://studio.aiagent-us1.cisco.com/static/assets/img/...
- AI engine ***: Webex AI Pro 1.0
- Welcome message ***: Hi there, how can I help you?

- The Profile tab on the Agent Configuration page, after creation, allows you to change the agent's image, the AI engine it uses, its goal, and the welcome message displayed when the AI agent is launched.
- These are the fields that control these aspects:
- URL for agent profile image**: The default URL from which the system fetches the AI agent's logo or image.
- AI engine**: The chosen AI engine is displayed.
- Welcome message**: The default welcome message that the AI agent uses to start the interaction.

 **Note**

By default, Webex AI Pro 1.0 (with GPT 4.0) is the available AI engine.

- Lets map the AI agent with the Knowledge base that was created.
- Navigate to Configurations > Knowledge tab and choose the configured knowledge base from the drop-down list.

AI agent configurations Draft

Change the agent configurations

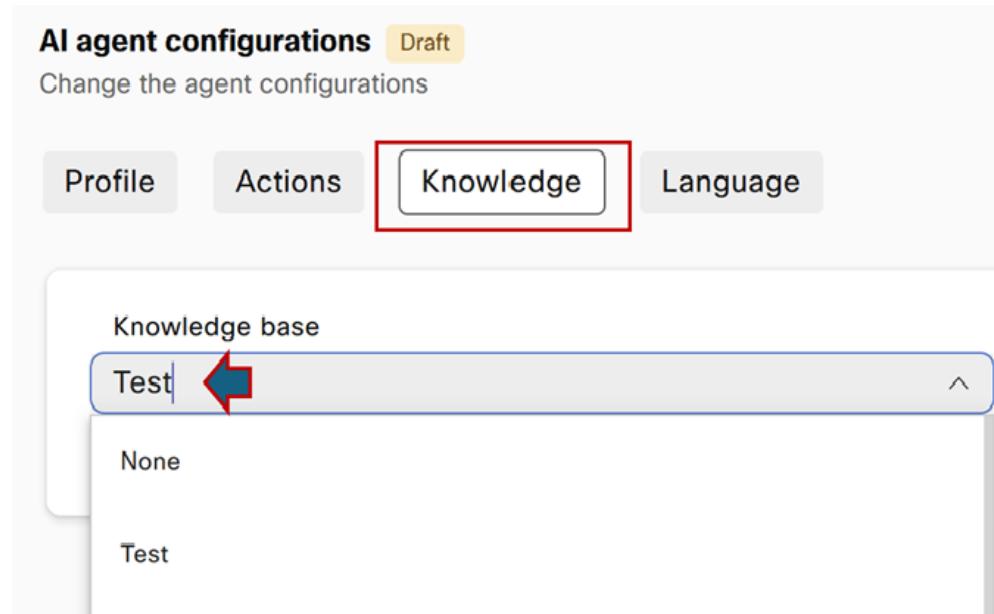
Profile Actions **Knowledge** Language

Knowledge base

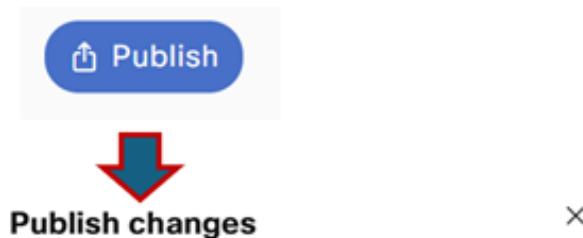
Test 

None

Test



- Click **Save changes** followed by **Publish** to make the AI agent live.



Name this version so that it can be tracked in History.

Version name *

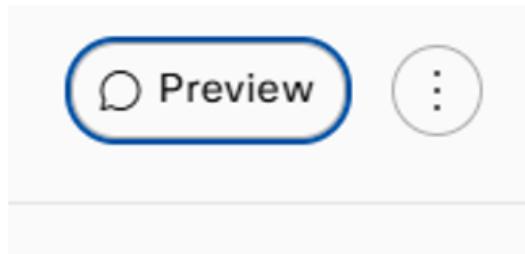
Version 1.0

Cancel

Publish

1.2.3 Section 2 : Test Webex AI Agent

- Now it's time to test the AI agent. Quick way to check if the agent is constructing and answering queries correctly is to use the Preview option.



- Click the "Preview" button , a chat window appears in the bottom right corner of the pane where questions can be entered.
Choose **Start a chat** option

A screenshot of a "Preview" window titled "Anuj_WebexOne_Car". The window contains the following text:

Start a call or chat to experience a simulated conversation with this AI agent. A voice transcript will show here for calls.

Use a headset for the best experience.

At the bottom of the window are two buttons: "Start a chat" (highlighted with a red box) and "Start a call".

- To test in this example, we pose the question "**What warranty comes with the vehicle?**"
The response provided was accurate and consistent with the knowledge base created for this AI agent.

The screenshot shows a Webex AI Agent interface. At the top, there's a purple circular icon with a white star-like symbol, followed by the text "Anuj_WebexOne_Car". Below this, a red banner on the left says "Question" and an arrow points right towards the message area. In the message area, a grey box contains the question "What warranty comes with the vehicle?". To the right of the question is the timestamp "5:05 PM". The main response is contained within a large grey box:

The warranty that comes with a vehicle depends on whether it is new or used:

- New cars come with a manufacturer's warranty, which covers specific repairs for a set period or mileage. Full details on coverage, including what is and isn't included, can be provided.
- Used cars may have an unexpired manufacturer's warranty that transfers to you, may be sold "as is," or may come with a dealer warranty. You will receive a clear written explanation of any warranty coverage for the specific vehicle you are interested in.

If you would like more details about a specific vehicle's warranty, please let me know!

At the bottom of the message area is a text input field with the placeholder "Type a message" and a send button with a blue arrow icon.

- The instruction section of the AI agent helps refine it to be more sophisticated and behave in the manner that the business dictates.

AI agent configuration Published

Add actions to connect your agent with external systems, or link a knowledge base to help it answer questions.

Profile Knowledge Actions Language

Agent name * Anuj_WebexOne_Car

System ID * Anuj_WebexOne_Car-8pGzpqCi

Agent's goal * ⓘ Answer Support Questions

Instructions ⓘ Enter step-by-step instructions for your agent to accomplish its goal.



Insert example

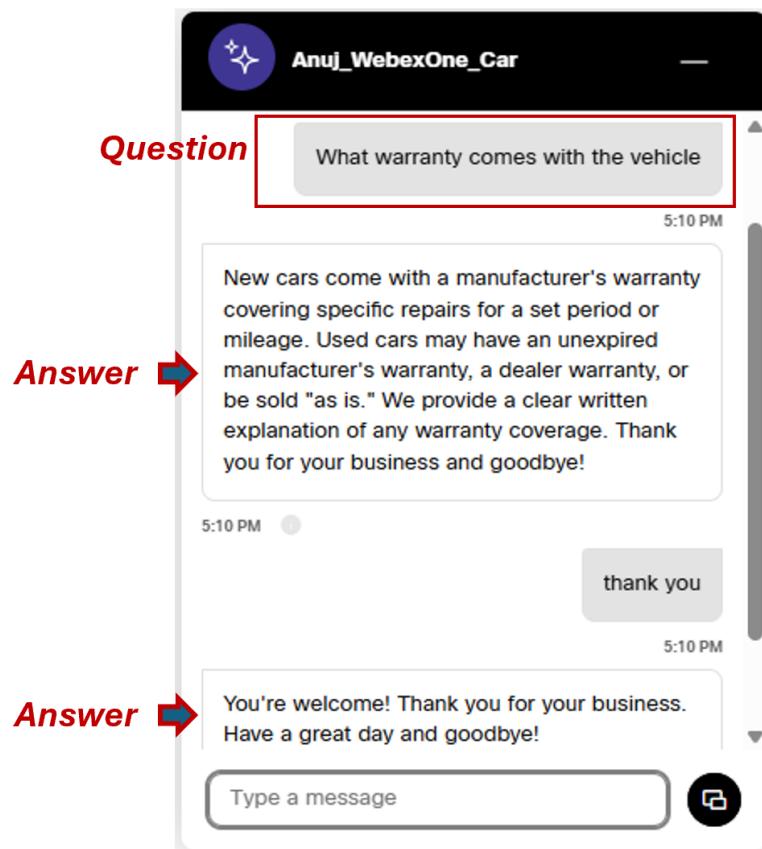
- In this example, let's work on these three areas of the bot with the following instructions:
- Tone and Demeanor:** Be polite, as the client is important.
- Role Definition:** Answer the question in no more than 50 words.
- Response Guidelines:** Thank the user for their interest and business.
- Add these guidelines to the Instructions section, save, and publish the changes.

Instructions ⓘ

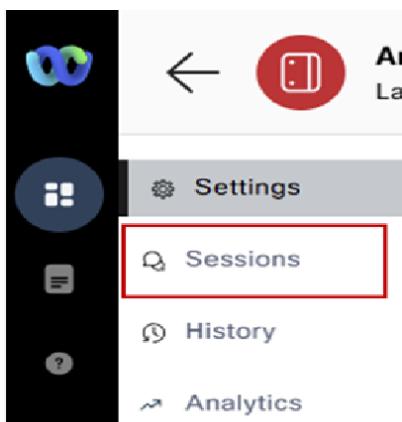
- Be polite as the client is important.
 - Answer the question in not more than 50 words.
 - Thank the user for their business and say goodbye.

Insert example

- To test, pose the same question: "**What warranty comes with the vehicle?**"
- As the Role Definition dictated answering within 50 words, you will see the answer within that limit.



- You can now view the details of these sessions and the history of configuration changes performed on the AI agent through the Sessions section.
- This information is useful for auditing, analyzing, and improving the AI agent.
- From the left navigation pane, click Sessions.



- The Sessions page appears, where each session is displayed as a record containing all the messages of the session, along with the following fields:

Channels	Session ID	Messages	Updated at	Metadata
1	95a59f1c-b56f-4e1c-a172-7db885228e8d	1	3 Feb 25, 1:42 PM	X
1	fe990d79-9b9e-4bc0-b92d-38f65f812011	1	27 Jan 25, 1:01 PM	X
3	35cb9bf4-5a81-4142-a39f-18d211e9c4c0	3	22 Jan 25, 9:24 AM	X
3	e641d562-61a2-4730-b3d7-88b4d4a69c24	3	22 Jan 25, 9:15 AM	X

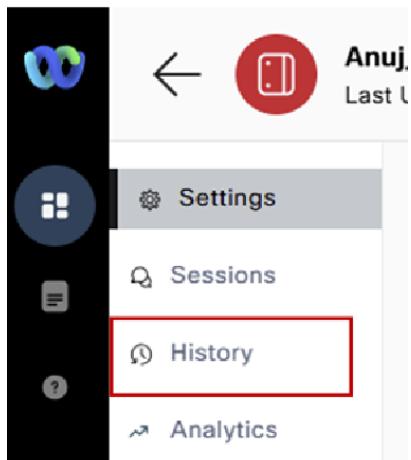
- **Channels:** The channel where the interaction took place.
- **Session ID:** The unique room ID or session ID for a conversation.
- **Room Metadata:** Additional information about the room.
- Click on an individual row in the sessions table for a detailed view of that session.
- The system displays the following session details:

The screenshot shows the Webex AI Agent interface. At the top, there's a table of session details with columns for Channels, Session ID, Messages, Updated at, and Metadata. Below the table, a message card shows a session ID and consumer ID. The main area is divided into two panes: the Left Pane and the Right Pane.

Left Pane: Displays transaction details. It includes a search bar, a Txn ID, and a list of messages. The messages show a customer asking about vehicle warranty and an AI agent providing a detailed response.

Right Pane: Titled "Right Pane" with an "Expand all" button. It shows "Knowledge utilized" and a specific entry for "BuyingCARFAQ.pdf". The entry contains a detailed explanation of warranty coverage, manufacturer warranties, and dealer warranties, along with a note about Cisco Confidential.

- The left panel displays details about the transactions.
- The right panel displays details about slot filling and fulfillment related to all actions. This is applicable to actionable actions for the bot.
- Finally, to view the details of configuration changes performed on the AI agent, you can use the History tab in the navigation pane.



- The History pane provides information about the version history and change logs.
- Please review this information at your convenience.

1.3 Result

- Congratulations on finishing the lab! Now, let's dive into the next exercise to enhance this agent so it can perform actions.

1.4 Task 2 - Build Autonomous Webex AI Agent Actions

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

Control Hub URL	https://admin.webex.com
Username	labuserID@wx1.wbx.ai (where ID is your assigned pod number; this ID will be provided by your proctor)
Password	webexONE1!

1.4.1 Objective

In this lab, audience will learn to explore and set up Autonomous Webex AI Agent actions using Webex Connect flows. They will also learn how to configure these flows to interact with external APIs, allowing the AI Agent to handle data and fulfill requests by connecting to other services.

1.4.2 Section 1 : Create a Webex Connect Flow for Car Order

- Action fulfillment for the agent is always handled via Webex Connect flows.
- Lets begin by creating a Webex Connect flow, which can later be selected to a specific Webex AI action.
- Navigate to the Webex Connect portal through the '**Quick Links**' section in the Webex Contact Center area of Control Hub.

Quick Links

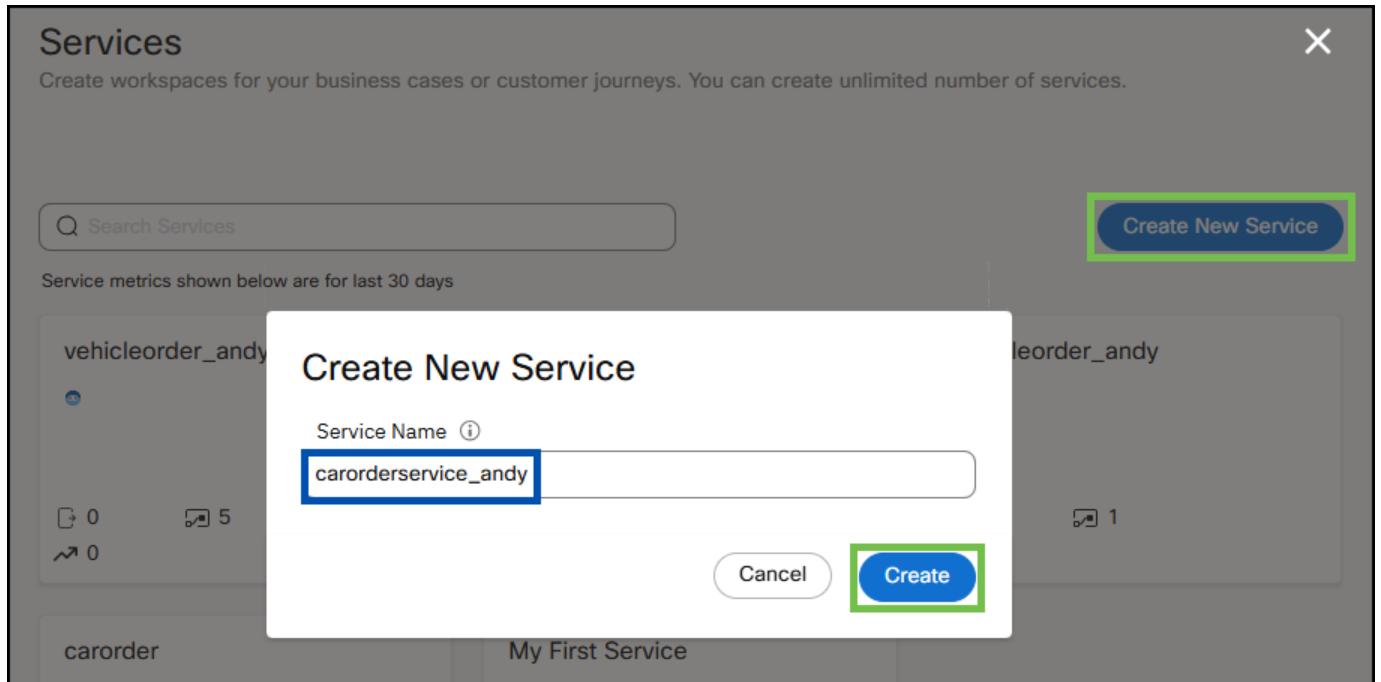
Contact Center Suite

- ⊕ [Desktop](#)
- ⊕ [Analyzer](#)
- ⊕ [Create new flow](#)
- ⊕ [Webex Contact Center Management Portal](#)
- ⊕ [Topic Analytics](#)
- ⊕ [Webex AI Agent](#)

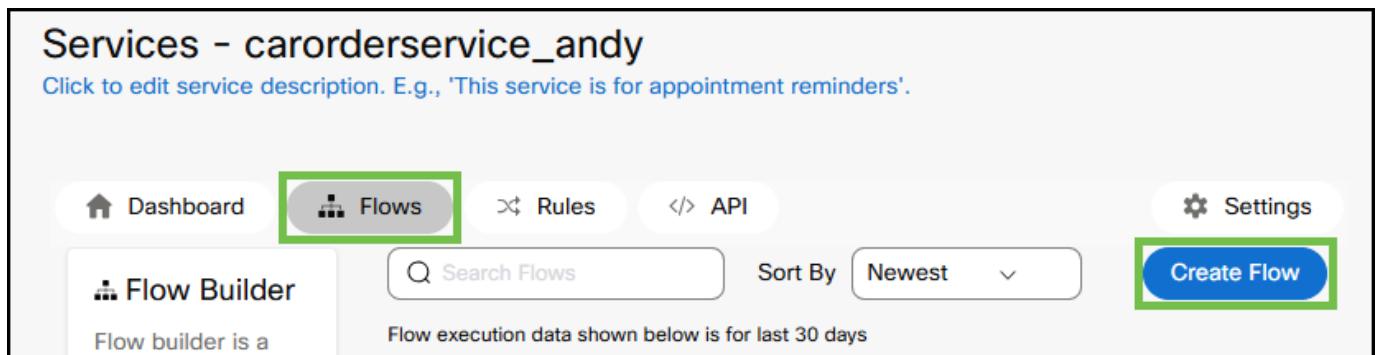
Digital Channels

- ⊕ [Webex Connect ⓘ](#)
- ⊕ [Webex Engage](#)

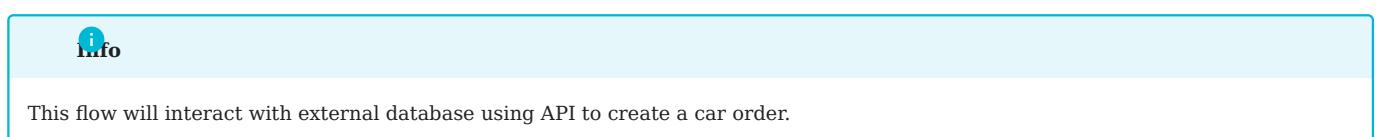
- Create a new service by clicking '**Create New Service**' and providing a name **Webexone_carorderservice_YourName_**.



- Navigate to 'Flows' section and click 'Create Flow'.



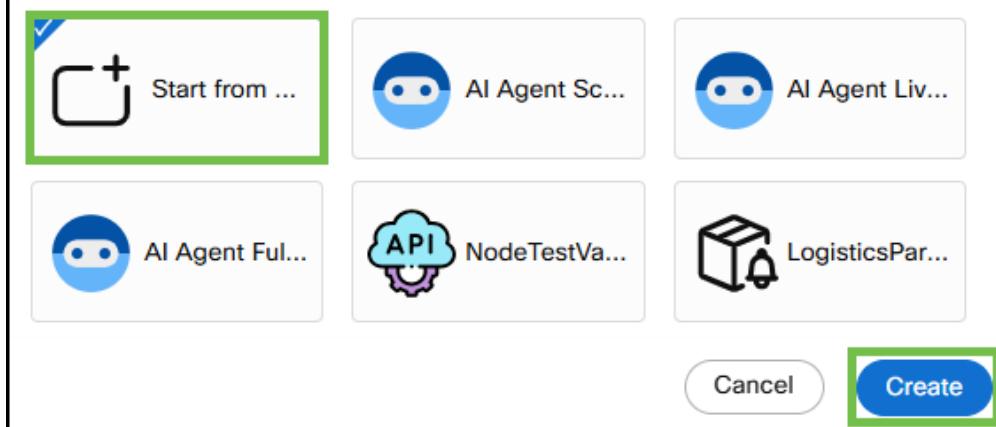
- Provide a 'Flow Name' (i.e. `carcreateorder_name`), ensure that 'Start from Scrach' is selected, and click 'Create'.



Create Flow

Flow Name
carordercreate

Method
New Flow



Start from ...

AI Agent Scenario

AI Agent Live

AI Agent Fulfillment

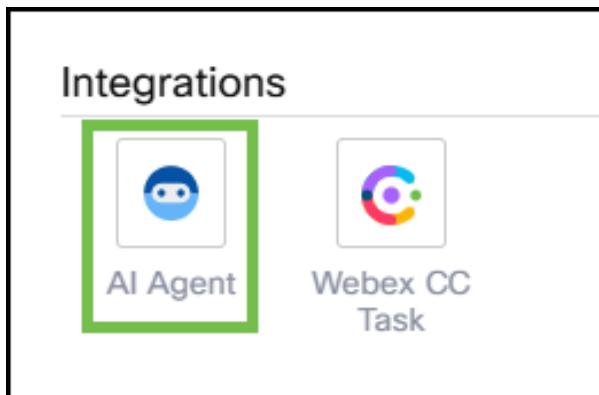
NodeTestVa...

LogisticsPar...

Cancel

Create

- In the next menu, select 'AI Agent' under 'Integrations'.



- 'Configure AI Agent Event' is displayed. Under 'Provide sample Json' provide the variables that will be passed from the Webex AI Agent. For a new car order, use the following:

```
{
  "model": "",
  "color": "",
  "engine": "",
  "customer": ""
}
```

- Click 'Parse' and 'Save'.

Configure AI Agent Event

Configuration **Transition Actions (Optional)** **Data Stream (Optional)** **Help**

Trigger from AI Agent to initiate flow

End Point URL ⓘ
https://hooks.us.webexconnect.io/flows

PROVIDE SAMPLE JSON ⓘ

```
1: {
2:   "model": "",
3:   "color": "",
4:   "engine": "",
5:   "customer": ""
6: }
```

OUTPUT VARIABLES ⓘ

PARSED VARIABLES(4)

color:
engine:
model:
customer:

Parse

Conditions
Flow will invoke only when these conditions are met

Start
Node ID: 2

Cancel **Save**

- Drag the '**HTTP Request**' node from the left-hand side menu, place it next to the '**AI Agent**' node, and connect them by clicking and holding the green dot on the '**AI Agent**' node and dragging it to the '**HTTP Request**' node.

carordercreate
Working Draft ▾

Build Analyse

Utilities Channels Integrations Q

UTILITIES

- Evaluate**
Execute your own JavaScript code for handling advanced business logic.
- Branch**
Branch flow execution based on conditional logic.
- HTTP Request**
Make an HTTPS request to the server and process the response.
- Delay**
Pause the flow for configured duration or until a fixed time and date.

```

graph LR
    A[Configure AI Agent Event] -- "onBegin" --> B[HTTP Request]
    
```

- Double click on the '**HTTP Request**' node and fill the fields as follows:

Field name	Value
Method	PUT
Endpoint URL	https://68085a3f942707d722ddf6a8.mockapi.io/api/cars
Header	Content-Type
Value	application/json
Output Variable Name (click '+Add Variable')	orderid
Response Entity	Body
Response Path	\$.orderid

For '**Body**' section use:

```
{
  "color": "$(n2.aiAgent.color)",
  "engine": "$(n2.aiAgent.engine)",
  "model": "$(n2.aiAgent.model)",
  "customer": "$(n2.aiAgent.customer)"
}
```

- Click '**Save**'.

HTTP Request

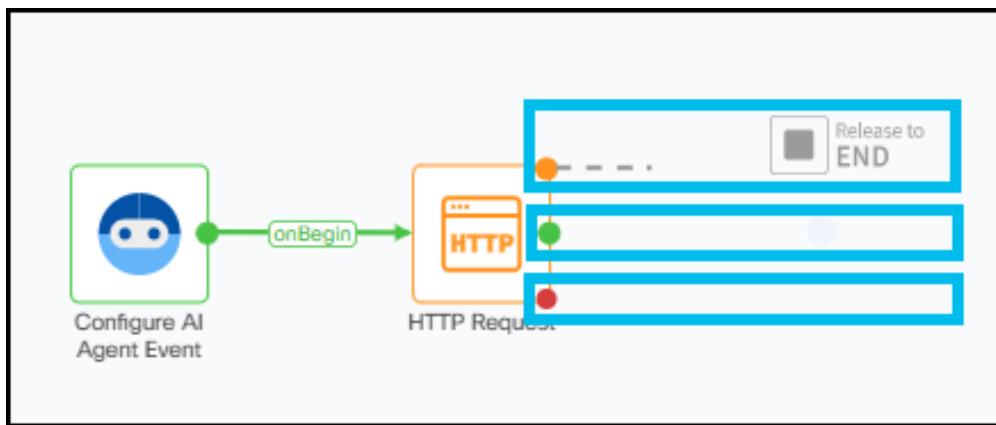
Help

Configuration **Transition Actions (Optional)**

Make GET, POST, PUT, PATCH and DELETE requests to your application

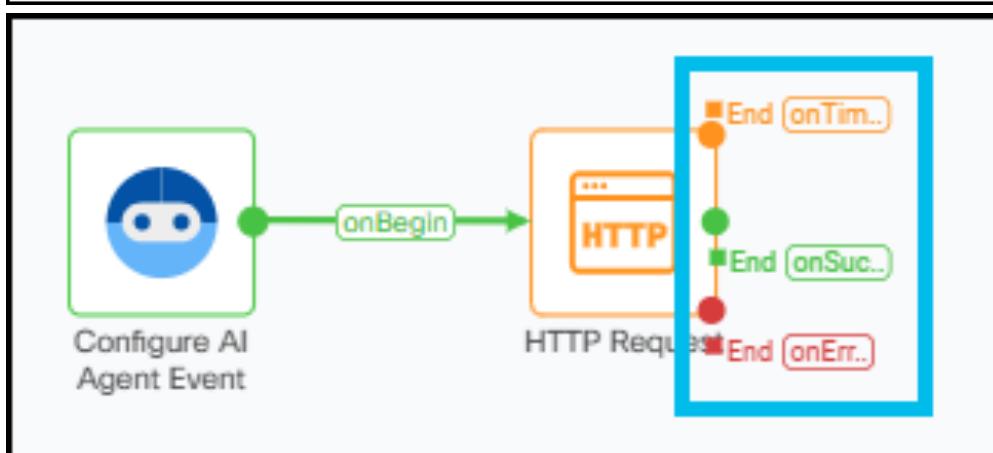
Method	Endpoint URL	
PUT	https://68085a3f942707d722ddf6a8.mockapi.io/api/cars	
Header	Value	
Content-Type	application/json	
+ Add Another Header		
Body		
<pre>{ "color": "\${n2.aiAgent.color}", "engine": "\${n2.aiAgent.engine}", "model": "\${n2.aiAgent.model}", "customer": "\${n2.aiAgent.customer}" }</pre>		
Connection Timeout ⓘ	Request Timeout ⓘ	
e.g., 1000 ms	e.g., 1000 ms	
Proxy Address (Optional) ⓘ		
e.g., 192.128.19.243:80		
Output Variables		
<input checked="" type="radio"/> JSON	<input type="radio"/> XML	Import From Sample
Output Variable Name ⓘ	Response Entity	Response Path ⓘ
orderid	Body	\$orderid
+ Add Variable		
HTTP Request Node ID: 3		Test Cancel Save

- Configure 'HTTP Request' node outcomes by dragging the orange, green, and red outcome dots to empty space on the canvas and selecting appropriate outcomes (e.g., 'onTimeout' - Error, 'onSuccess' - Success, 'onError' - Error).



End

Configuration	Transition Actions (Optional)
Map node events and flow results to end the flow	
Node Event	Flow Result
onTimeout ▾	102 - Flow completed with an error [Error] ▾
End Node ID: 4	
Cancel Save	



- Next, click '**Flow Settings**' (gear icon), navigate to '**Flow Outcomes**', and expand '**Last Execution Status**' outcome.
- Click '**Add New**', use 'orderid' under '**Key**' and '\$(n3.orderid)' under '**Value**'.
- Click '**Save**'.

Flow Settings

Flow Outcomes (highlighted)

Custom Variables

OUTCOME: Last Executive

NOTIFY: Enabled

Notification Settings

Notify AI Agent (radio button selected)

Define key-value pairs to be sent to the AI Agent.

KEY	VALUE
transactionID	\$(transid)
flowname	\$(flowname)
serviceName	\$(serviceName)
statuscode	1000
orderid	\$(n3.orderid)

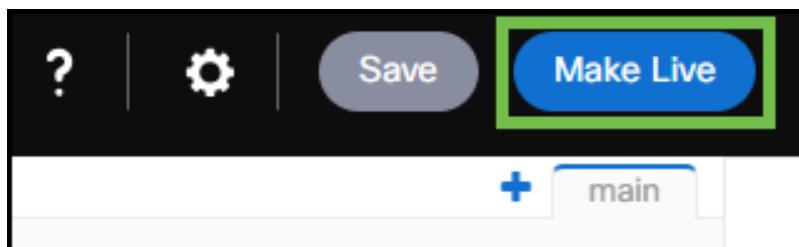
+ Add New

Flow id: 59121

Cancel

Save (highlighted)

- Make the flow live by clicking '**Make Live**' (top right corner) and then clicking '**Make Live**' again in the next dialog box.



1.4.3 Section 2 - Configure AI Agent Car Order Action

- Navigate back to the AI Agent Studio and to your Autonomous Webex AI Agent.
- Select '**Actions**' and click '**New Action**'.

AI agent configuration Ready to preview

Add actions to connect your agent with external systems, or link a knowledge base to help it answer questions.

Profile **Knowledge** **Actions** **Language**

?

+ New action

- Provide an '**Action Name**' (use *carordercreate*).
- Provide the following '**Action description**':

Submit a car order when the customer chooses a model and all other customization options.
Provide orderid after a successful order.

- Under '**Action scope**', select '**Slot filling and fulfillment**'.

General information

Action name *

carordercreate

Action description * ⓘ

Submit a car order when customer chooses model and all other customization options. Provide orderid after successful order.

Action scope *

Slot filling and fulfillment

- Select the service you created in the previous task under '**Select service**' and the flow created above under '**Select a flow**'.

Webex Connect Flow Builder Fulfillment

Determine how to handle customer requests or resolve issues.

[Learn more about Webex Connect Flow Builder for fulfillment](#)

Select service *

carorderservice_andy

Select a flow *

carordercreate

- Click 'New input entity'.

Slot filling

Add input entities in a table or define input parameters in JSON for slot filling to ensure the AI agent gathers all necessary information.

Input entities							Use JSON instead
Entity Name	Type	Value	Description	Example	Required	Controls	+ New input entity

- Add four entities for the Webex AI Agent to pass to the Webex Connect flow with the following parameters:

Entity name	Entity type	Entity description	Entity examples	Required
model	String	Car model that the customer chose	NovaTerra	Yes
color	String	Car color that the customer chose	Ruby Red	Yes
engine	String	Engine type that the customer chose	gas, electric	Yes
customer	String	Customer's name	Jack, Nick	Yes

Add a new input entity

Specify the details to collect from customers for more accurate AI responses

Entity name * ⓘ

Entity type

String

Entity description *

Car color that the customer chose

Entity examples ⓘ

Ruby Red

+ Add

Required *

Yes

No

Cancel

Add

Slot filling

Add input entities in a table or define input parameters in JSON for slot filling to ensure the AI agent gathers all necessary information.

Input entities

Use JSON instead ⓘ

Entity Name	Type	Value	Description	Example	Required	Controls
color	string	-	Car color that the customer chose	Ruby Red	Yes	 
customer	string	-	Customer's name	Jack, Nick	Yes	 
engine	string	-	Engine type that the customer chose	gas, electric	Yes	 
model	string	-	Car model that the customer chose	NovaTerra	Yes	 

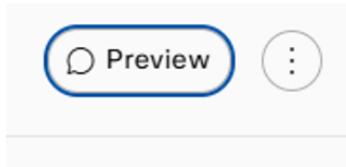
- Click 'Add' and Publish the changes.

X Cancel

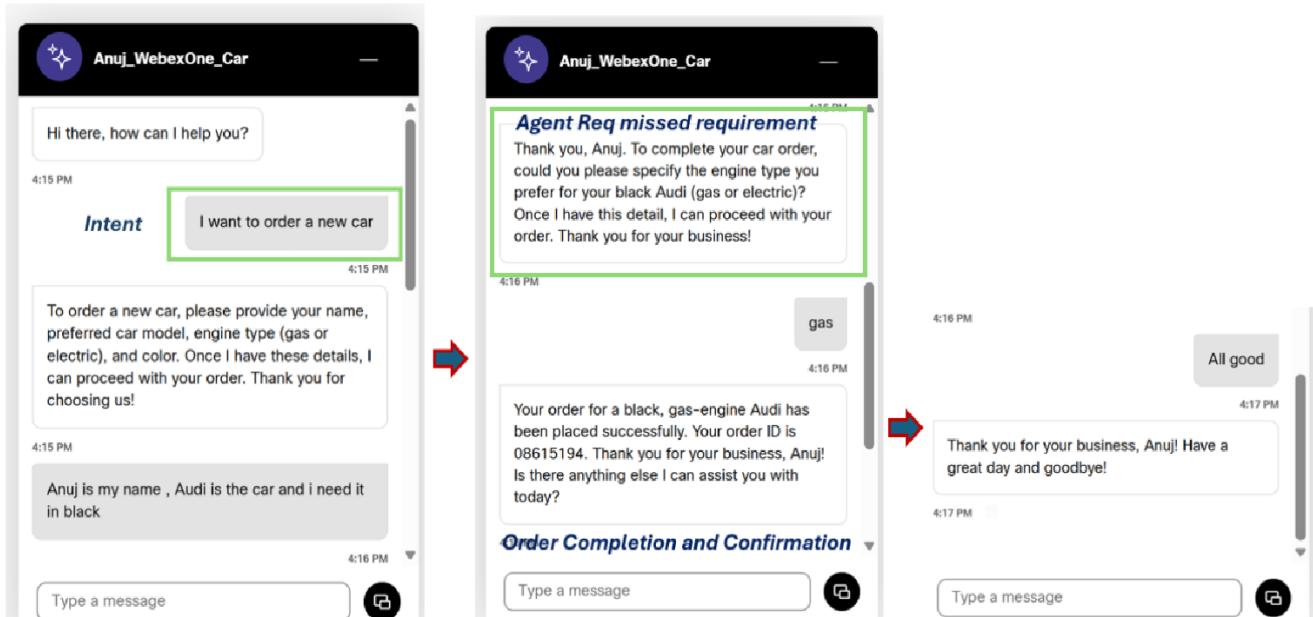
→ Add

1.4.4 Section 3 - Test Webex AI Agent

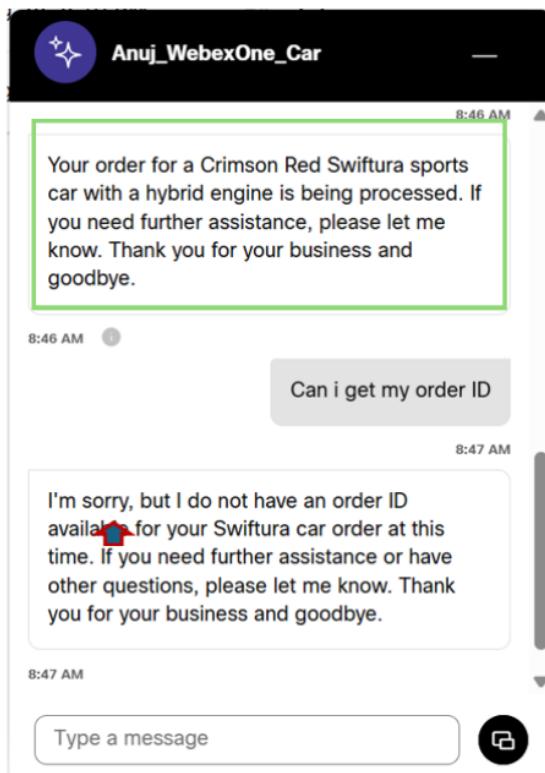
- Let's test the logic by creating a new car order by going back to AI Agent Studio and selecting the Autonomous AI Agent created in exercise 1.
- Click 'Preview' and Verify the functionality by creating a car order.



- For guidance the screenshot provides an example of how an order is created.
- First, the user expressed the intent to buy a new car.
- As per its instructions, the AI agent requested car details like the model, engine type, and color.
- When a detail was missed, the agent was able to ask the user to provide the missing parameter to complete the order.
- Once it received all the necessary information, it successfully created the order and provided the order details.



- In your case, you will see that the agent is not able to provide the order ID. Specifically asking the agent for the order ID was not helpful either.



- To begin, verify the session details to confirm the agent's actions by going into the Agent and Sessions Section.

Channels	Session ID
...	ea86dabf-277b-4bec-afc6-78f872fd5e38
...	8b2e5858-150b-4abf-bda2-9b7e3c4b4ad3
...	a588bf06-ece9-48cd-bcf2-923d50a883b7

- Choose the latest Session ID.
- Note that the fulfillment was successful, and information was provided to the Connect Flow for order execution.

Session ID: c6bb3648-c69b-47a2-bb3f-30bbb9b33bc7
Consumer ID: cb9e1481-6795-4732-9695-2054b8d7ef75 Total Messages: 16 Last updated at: Aug 15 '25 8:47 AM

Txn ID: 0350c722c0594ece14a67e886101920 8:46:11 AM, Aug 15, 2025

Switura 8:46:20 AM, Aug 15, 2025

Performing action "carordercreate"

Your order for a Crimson Red Swiftura sports car with a hybrid engine is being processed. If you need further assistance, please let me know. Thank you for your business and goodbye.

8:46:23 AM, Aug 15, 2025

Txn ID: d25dde30f0fe402fb4c7a6075ef3239e 8:47:13 AM, Aug 15, 2025

Can I get my order ID

I'm sorry, but I do not have an order ID available for your Swiftura car order at this time. If you need further assistance or have other questions, please let me know. Thank you for your business and goodbye.

8:47:15 AM, Aug 15, 2025

model: Swiftura

- Required: true
- Entity Type: string
- Entity description: Car model that the customer chose

Fulfillment: Success (0.9s)

Service and flow info:

- Service: Webexone_carorderservice_Anuj
- Flow: CarOrderAnuj

Fulfillment output:

- Transaction ID: 21debbf5-b390-4b76-b2c1-81291459b6c3
- Size: 161
- Output:

```

1 {
2   "flowname": "CarOrderAnuj",
3   "orderid": "",
4   "serviceName":
5     "Webexone_carorderservice_Anuj",
6   "statuscode": "1000",
7   "transactionID": "21debbf5-b390-4b76-b2c1-
81291459b6c3"
8 }
```

Successful Fulfillment sent to Flow for order execution

- Next, lets examine the Connect Flow debugs to see how the information was executed.
- To access these, navigate to Control Hub > Contact Center Quick Links > Webex Connect, and then select the Connect service you created.

Quick Links

- Contact Center Suite
- * Desktop
- * Analyzer
- * Create new flow
- * Webex Contact Center Management Portal
- * Topic Analytics
- * Webex AI Agent

Digital Channels

- * Webex Connect
- * Webex Engage
- * Manage proactive chats

Services

Create workspaces for your business cases or customer journeys. You can create unlimited workspaces.

Search Services

Service metrics shown below are for last 30 days

Webexone_carorderservice_Anuj

0 7 0

- Within the service, select your created Flow and click the Debug option on the right side. The Transaction Logs window will open.

The screenshot shows the Flow Builder interface. At the top, there are tabs for Dashboard, Flows (selected), Rules, and API. Below the tabs is a search bar labeled "Search Flows" and a dropdown for "Sort By" set to "Newest". A message states "Flow execution data shown below is for last 30 days". A table lists flows, with one row highlighted: "Trigger" (Configure AI Agent Event), "Flow" (CarOrderAnuj, Flow Id:67269), "Status" (Live), and "Exe" (7). A blue toggle switch is next to the status. A red arrow points down from the flow table to the flow diagram below.

Flow Builder

Flow builder is a visual designer with a library of nodes that you can drag and drop to create and test interactive customer journeys with little to no code.

[Check out our guide to building flows here.](#)

If you've got more questions after reading through it, just ask. We're always here to help.

[Create Flow](#)

```

graph LR
    A[Configure AI Agent Event] -- "onBegin" --> B[HTTP Request]
    B -- "End onTim." --> C(( ))
    B -- "End onSuc." --> D(( ))
    B -- "End onErr." --> E(( ))

```

- First, click Decrypt Logs to convert the logs into text, then select the latest transaction.

The screenshot shows the Transaction Logs interface. At the top, there's a header with "TRANSACTION LOGS" and a search bar with "Aug 15th, 2025 00:00:00 to Aug 15th, 2025 23:59", "Transaction ID", and a "SEARCH" button. Below the header is a table with columns: TIME STAMP, TRANSACTION ID, INVOKED BY, TIME TAKEN (HH:mm:ss.SSS), and LAST NODE. Two transactions are listed:

TIME STAMP	TRANSACTION ID	INVOKED BY	TIME TAKEN (HH:mm:ss.SSS)	LAST NODE
15-08-2025T13:46:22.467+01:00	21debbf5-b390-4b76-b2c1-81291459b6c3	Invoked by API	00:00:00.285	End - Success
15-08-2025T13:38:17.548+01:00	68bd8840-2c72-421f-97ea-88a5ab9f569b	Invoked by API	00:00:00.301	End - Success

A green box highlights the transaction ID "21debbf5-b390-4b76-b2c1-81291459b6c3". Above the table, there are buttons for "DECRYPT LOGS", "HELP", and "C".

- You will see three nodes being executed.
- The "Configure AI Agent Event" payload is constructed and passed to the "HTTP Request Node."

TRANSACTION LOGS 21debbf5-b390-4b76-b2c1-81291459b6c3 X

Trans ID : 21debbf5-b390-4b76-b2c1-81291459b6c3

NODE ID	NODE	OUTCOME	TIME TAKEN(HH:mm:ss.)
2	Configure AI Agent Event	onBegin	00:00:00 . 000

↓

Node Trans ID : 0b6ee3ec-e120-4de4-8401-d6d80e074a0e

Transaction Details

Transaction details for AI Agent event is available for a maximum of 7 days.

Payload :

```
{
    "color": "Crimson Red",
    "engine": "Hybrid",
    "model": "Swiftura"
}
```

Timestamp : 15-08-2025T13:46:22.000+01:00

□

□

□

- In the HTTP Request Node, you'll see the request data was constructed and the API was executed. However, the response data shows a status of 400 Bad Request, indicating the request was not processed due to an invalid request.

TRANSACTION LOGS 21debbf5-b390-4b76-b2c1-81291459b6c3 X

Trans ID : 21debbf5-b390-4b76-b2c1-81291459b6c3

NODE ID	NODE	OUTCOME	TIME TAKEN(HH:mm:ss.SSS)
2	Configure AI Agent Event	onBegin	00:00:00 . 000
3	HTTP Request	oncomplete	00:00:00 . 283

↓

Transaction Details

Request Times : 15-08-2025T13:46:22.184+01:00

Request URL : <https://68085a3f942707d722ddf6a8.mockapi.io/api/cars>

Request Data :

```
{
    "url": "https://68085a3f942707d722ddf6a8.mockapi.io/api/cars",
    "method": "PUT",
    "headers": {
        "x-wx-gtrid": "de4912f0-30be-5daa-b99c-06882a4e9161",
        "Content-Type": "application/json"
    },
    "payload": "{\n        \"color\": \"Crimson Red\",\n        \"engine\": \"Hybrid\",\n        \"model\": \"Swiftura\"\n    }\n    {\n        \"customer\": \"\"\n    }"
}
```

exec

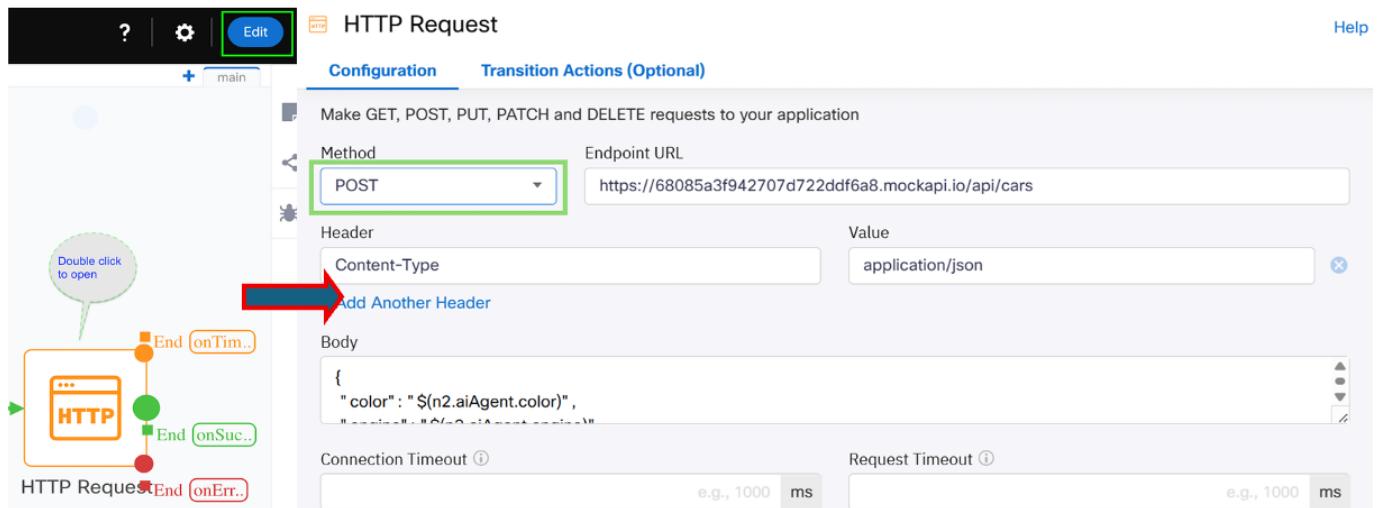
Response Data :

```
{
    "headers": {
        "Server": "Heroku",
        "Nel": "{"report_to": "heroku-nef", "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1},
        "Access-Control-Allow-Origin": "*",
        "Access-Control-Allow-Methods": "GET,PUT,PATCH,DELETE,OPTIONS",
        "Date": "Tue, 22 Aug 2025 13:46:22 GMT",
        "Via": "1.1 heroku-router",
        "Content-Type": "application/json"
    },
    "body": "{\"msg\":\"Invalid request\"}",
    "status": 400
}
```

Failure

Request Executed via PUT

- The issue lies in the method selected to create the record. PUT is generally used for updating or replacing existing content, but for creating a new record, the correct method is POST.
- To correct this, select the Edit option in the flow. Then double click on the HTTP Request Node and change the Method from PUT to POST.



- Save the flow and make it Live.
- Create a car order again from Preview chat, and the execution should now be successful with AI agent returning an Order ID.

Wx1-DKauto007

I would like a nuclear powered Swiftura in Forest Green

5:36 PM

Thank you, DK Singh. The Swiftura is not available in Forest Green or with a nuclear engine. Please select Midnight Blue, Pearl White, or Crimson Red, and choose Hybrid or Electric engine to proceed. Thank you for your business and goodbye.

5:36 PM

ok can I order Midnight blue electric then

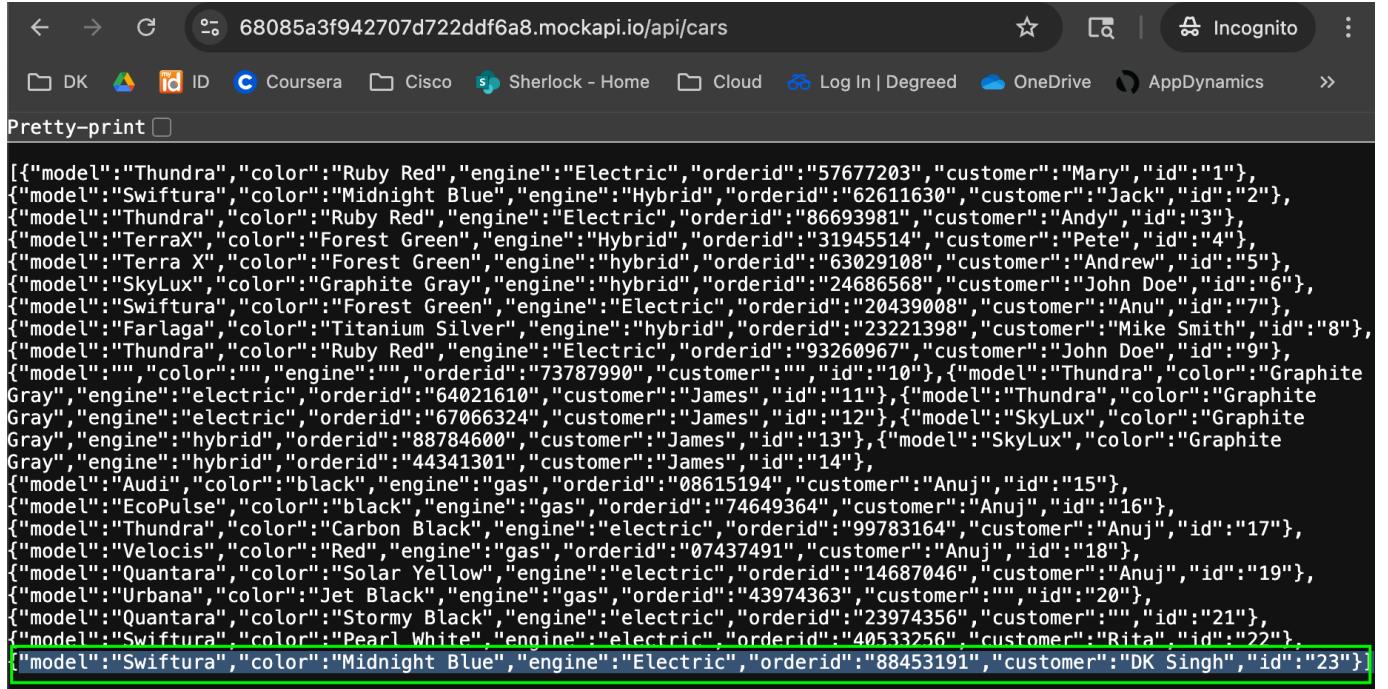
5:37 PM

Your order for a Midnight Blue, Electric Swiftura has been placed successfully. Your order ID is 88453191. Thank you for your

Type a message

- You can also access the datastore where these orders are stored and updated by entering the following URL into your browser:

Car Orders



```
[{"model": "Thundra", "color": "Ruby Red", "engine": "Electric", "orderid": "57677203", "customer": "Mary", "id": "1"}, {"model": "Swiftura", "color": "Midnight Blue", "engine": "Hybrid", "orderid": "62611630", "customer": "Jack", "id": "2"}, {"model": "Thundra", "color": "Ruby Red", "engine": "Electric", "orderid": "86693981", "customer": "Andy", "id": "3"}, {"model": "TerraX", "color": "Forest Green", "engine": "Hybrid", "orderid": "31945514", "customer": "Pete", "id": "4"}, {"model": "Terra X", "color": "Forest Green", "engine": "hybrid", "orderid": "63029108", "customer": "Andrew", "id": "5"}, {"model": "SkyLux", "color": "Graphite Gray", "engine": "hybrid", "orderid": "24686568", "customer": "John Doe", "id": "6"}, {"model": "Swiftura", "color": "Forest Green", "engine": "Electric", "orderid": "20439008", "customer": "Anu", "id": "7"}, {"model": "Farлага", "color": "Titanium Silver", "engine": "hybrid", "orderid": "23221398", "customer": "Mike Smith", "id": "8"}, {"model": "Thundra", "color": "Ruby Red", "engine": "Electric", "orderid": "93260967", "customer": "John Doe", "id": "9"}, {"model": "", "color": "", "engine": "", "orderid": "73787990", "customer": "", "id": "10"}, {"model": "Thundra", "color": "Graphite Gray", "engine": "electric", "orderid": "64021610", "customer": "James", "id": "11"}, {"model": "Thundra", "color": "Graphite Gray", "engine": "electric", "orderid": "67066324", "customer": "James", "id": "12"}, {"model": "SkyLux", "color": "Graphite Gray", "engine": "hybrid", "orderid": "88784600", "customer": "James", "id": "13"}, {"model": "SkyLux", "color": "Graphite Gray", "engine": "hybrid", "orderid": "44341301", "customer": "James", "id": "14"}, {"model": "Audi", "color": "black", "engine": "gas", "orderid": "08615194", "customer": "Anuj", "id": "15"}, {"model": "EcoPulse", "color": "black", "engine": "gas", "orderid": "74649364", "customer": "Anuj", "id": "16"}, {"model": "Thundra", "color": "Carbon Black", "engine": "electric", "orderid": "99783164", "customer": "Anuj", "id": "17"}, {"model": "Velocis", "color": "Red", "engine": "gas", "orderid": "07437491", "customer": "Anuj", "id": "18"}, {"model": "Quantara", "color": "Solar Yellow", "engine": "electric", "orderid": "14687046", "customer": "Anuj", "id": "19"}, {"model": "Urbana", "color": "Jet Black", "engine": "gas", "orderid": "43974363", "customer": "", "id": "20"}, {"model": "Quantara", "color": "Stormy Black", "engine": "electric", "orderid": "23974356", "customer": "", "id": "21"}, {"model": "Swiftura", "color": "Pearl White", "engine": "electric", "orderid": "40533256", "customer": "Rita", "id": "22"}, {"model": "Swiftura", "color": "Midnight Blue", "engine": "Electric", "orderid": "88453191", "customer": "DK Singh", "id": "23"}]
```

Congratulations! - You have successfully completed this task and the lab! If you'd like, you can also explore Bonus exercise 5 to dive and see how the orders can be modified.

1.5 Lab 3 - Create and Test Voice Flows for Webex AI Agents

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

Control Hub URL	https://admin.webex.com
Username	labuserID@wx1.wbx.ai (where ID is your assigned pod number; this ID will be provided by your proctor)
Password	webexONE1!

1.5.1 Objective

This lab exercise aims to guide participants in creating and configuring a voice flow for an Autonomous Webex AI Agent that was set up in a previous exercise. Participants will then interact with the AI Agent via a phone call to test the voice flow.

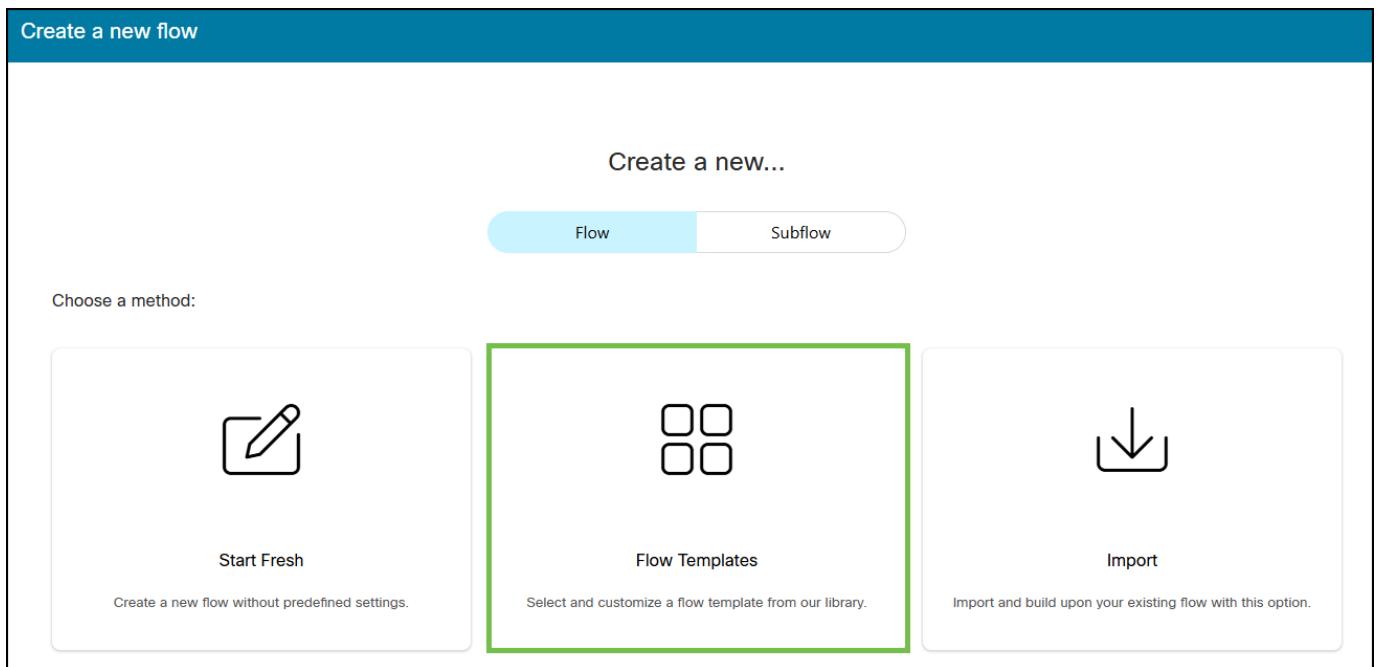
This exercise will also provide an opportunity to experience the AI Assistant feature of the contact center by highlighting the virtual agent's conversation summary when the call is escalated to a live agent

1.5.2 Section 1 - Voice flow configuration

- Navigate to the Control Hub and log in using the credentials provided above.
- After logging in to the Control Hub, navigate to the '**Flows**' menu on the-left hand side.
- Click '**Manage Flows**' and select '**Create Flows**'.

Flow	Description	Status	Last edited by	Last modified	Import Flows		
ai_agent_autono...	To learn how to config...	D...	admwxccciscoliv...	April 30, 2025			
ai_agent_autono...	To learn how to config...	D...	admwxccciscoliv...	April 30, 2025 11...			
testvariable		D...	admwxccciscoliv...	April 30, 2025 11...			
HelloWorld_Tem...	To learn how to config...	P...	admwxccciscoliv...	April 23, 2025 0...			
ai_agent_autono...	To learn how to config...	P...	labuser1@wxccc...	May 01, 2025 16:...			
ai_agent_autono...	To learn how to config...	P...	admwxccciscoliv...	April 30, 2025 11...			
ai_agent_autono...	To learn how to config...	P...	admwxccciscoliv...	April 30, 2025 11...			

- When prompted to '**Choose a method**', select '**Flow Templates**'.



- Choose '**AI Agent Autonomous (Package Tracking)**', click '**Next**', provide a flow name (e.g., '*ai_agent_autonomous_andy*'), and click '**Create Flow**'.

Create a new flow



Choose a template Name & Configure

Choose a template

Browse our collection of customizable flow templates to find the perfect fit for your needs.

[Learn more.](#)

autonomous



AI Agent Autonomous (Package Tracking)

by Cisco

This flow utilizes an **autonomous AI** agent to manage voice interactions related to package tracking. The flow provides an option to escalate to human agents when necessary or on AI Agen...

[View Details](#)

Back

Next

Create a new flow



Choose a template Name & Configure

Name & Configure

Template 'AI Agent Autonomous (Package Tracking)' selected. Name and configure your Flow.

Flow Name

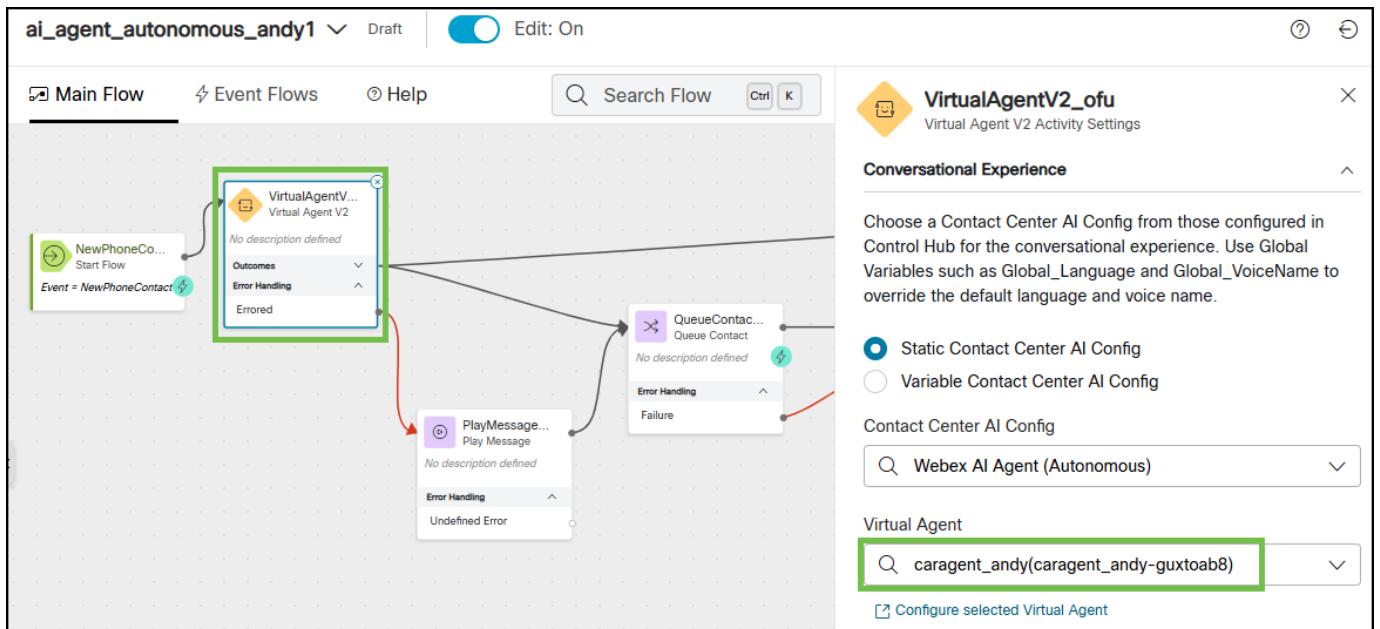
ai_agent_autonomous_andy1

Flow Name must be unique with no spaces. It can only contain alphanumerics and underscores.

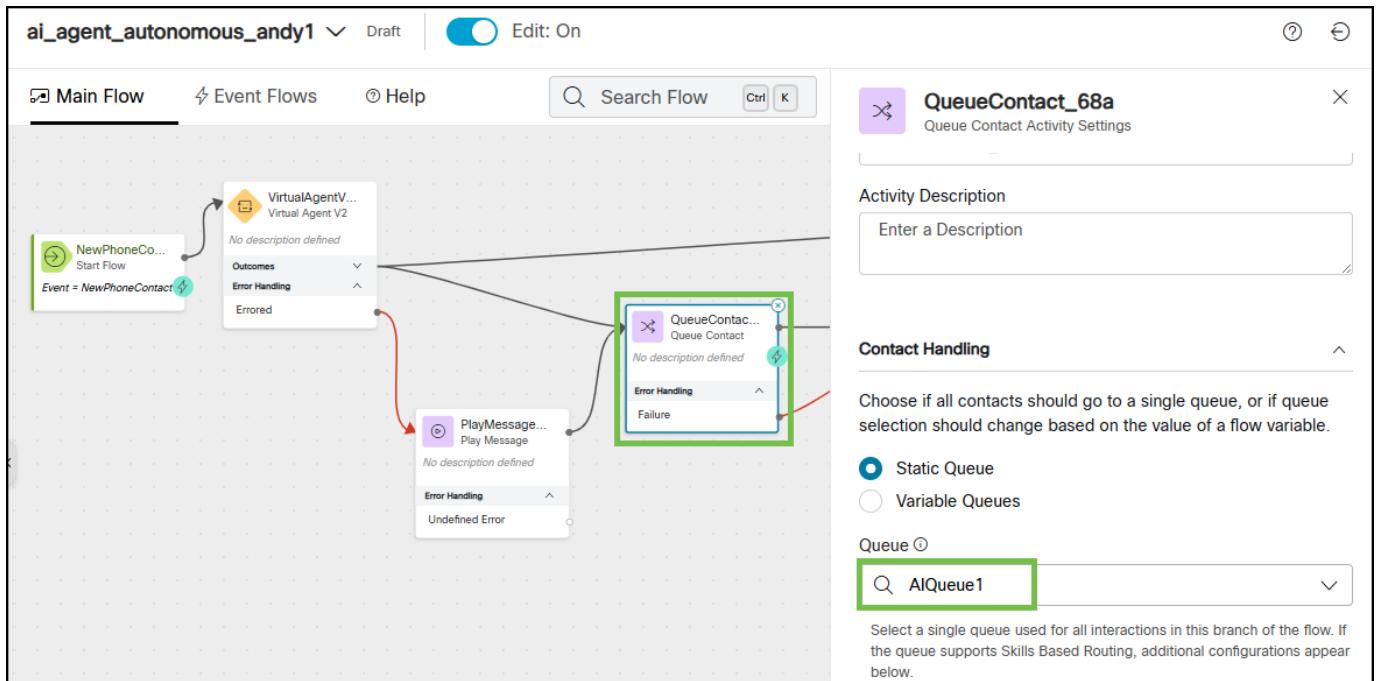
Back

Create Flow

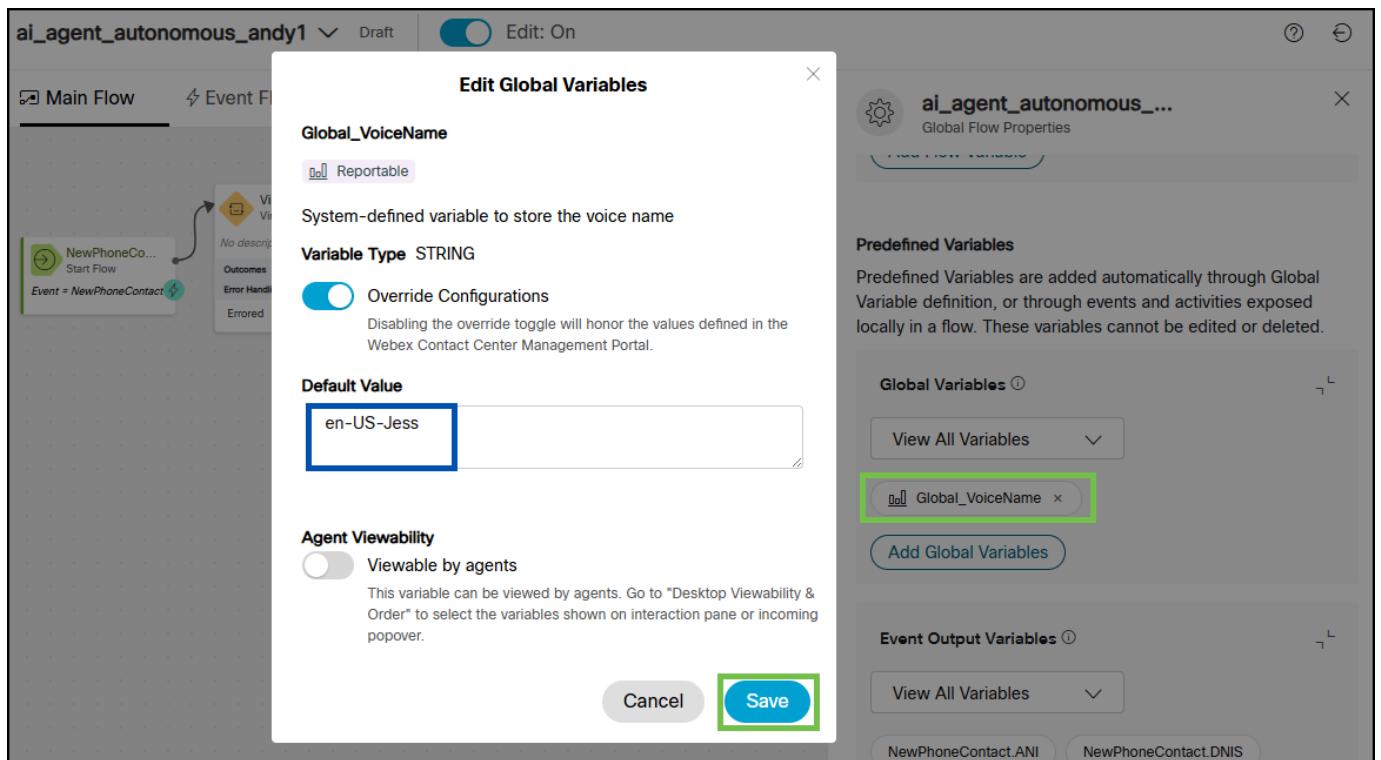
- Once the flow loads, click on the '**VirtualAgentV2**' node and select the Webex AI Agent (created in **Excercise 1**) under '**Virtual Agent**'.



- Click on the '**QueueContact**' node and select '**WebexOne_AIQueue_N**' (where '**N**' is your lab user number).



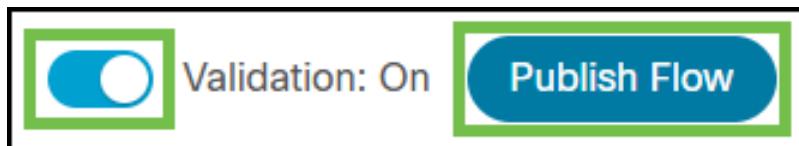
- Click on an empty space in the flow, then on the right-hand side, navigate to '**Global Variables**'. Click on '**Global_VoiceName**', select edit option (pencil icon), use 'en-US-Jess' for '**Default Value**', and click '**Save**'.



Note

Instead of Jess as the voice, you can choose different voice options like ***'en-US-Maria'** and ***'en-US-Henry'***. Supported voice languages for AI agents can be checked via Supported Languages and Voices for AI agents.

- Turn Flow Validation 'On' by clicking the 'Validation' button at the bottom of the page to publish the flow. Once validation is complete, click 'Publish Flow' and then 'Publish Flow' again in the next dialog box (**Latest** version label is selected automatically).



- Navigate to 'Channels' menu in the Webex Contact Center configuration.
- Open 'WebexOne_AI_EntryPoint_N' (_ where 'N' is your lab user number), associate your created flow under 'Routing Flow' (note down the 'Support Number' associated with this channel - it will be needed later in this lab for testing), and click 'Save'.

CUSTOMER EXPERIENCE

Channels			
Queues	WebexOne_AI_EntryPoint_2	AI_EP_2	Inbound Telephony
Business Hours	WebexOne_AI_EntryPoint_3	AI_EP_3	Inbound Telephony
Audio Files	WebexOne_AI_EntryPoint_4	AI_EP_4	Inbound Telephony
Flows	WebexOne_AI_EntryPoint_5	AI_EP_5	Inbound Telephony
AI Agents			
Call Recording Sc			
Functions			
Surveys			

AIChannel1

Entry Point Settings		Service Level Threshold ⓘ *	60	Seconds
Timezone (Business Hours only) *	America/New_York			
Routing Flow	ai_agent_autonomous_andy1			
Version Label	Latest			
Music on Hold	defaultmusic_on_hold.wav			

Support Number

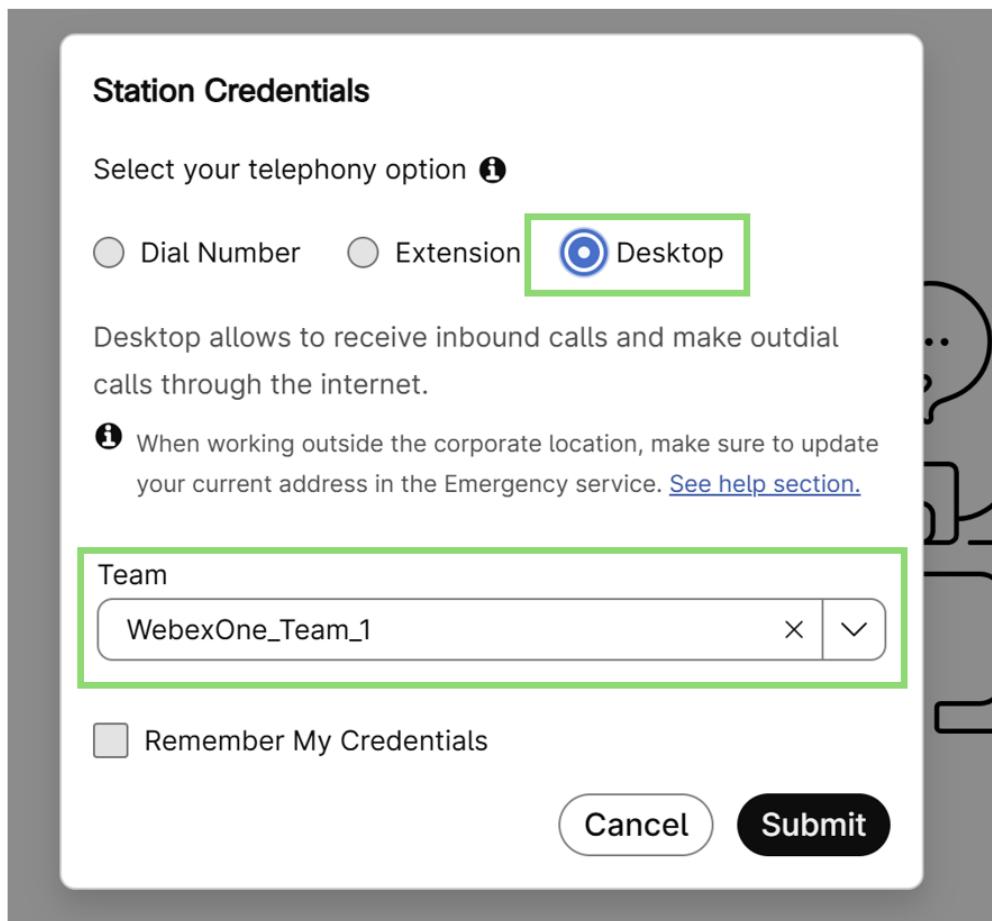
Your customers will use these numbers to reach your Contact Center. You can pick more than one number from the list of numbers available and already set up in Webex Calling services.

Number	Webex Calling location	Support Number	PSTN Region	Actions
1	357c9a4a-4be5-41f5-b273-4b77adc8ca09	+13503051030	Default	

Cancel **Save**

1.5.3 Section 2 - Verifying Voice Flow, Agent Handover and AI Assistant Summary

- Log in to the Webex Contact Center Agent Desktop:
- For the phone number, select '**Desktop**'
- For the team use **WebexOne_Team_N** (where 'N' is your lab user number').



- Call the channel number (from the steps above), interact with the Webex AI Agent and order the car and get the order ID.
- When you interact with the AI agent, you might notice a couple of issues.
- First, the bot may not be able to answer. This could be due to the agent's AI engine. If you notice this issue, you can change the AI engine to Webex AI Pro-Us 1.0 in the configured AI Agent Profile and test the call again.

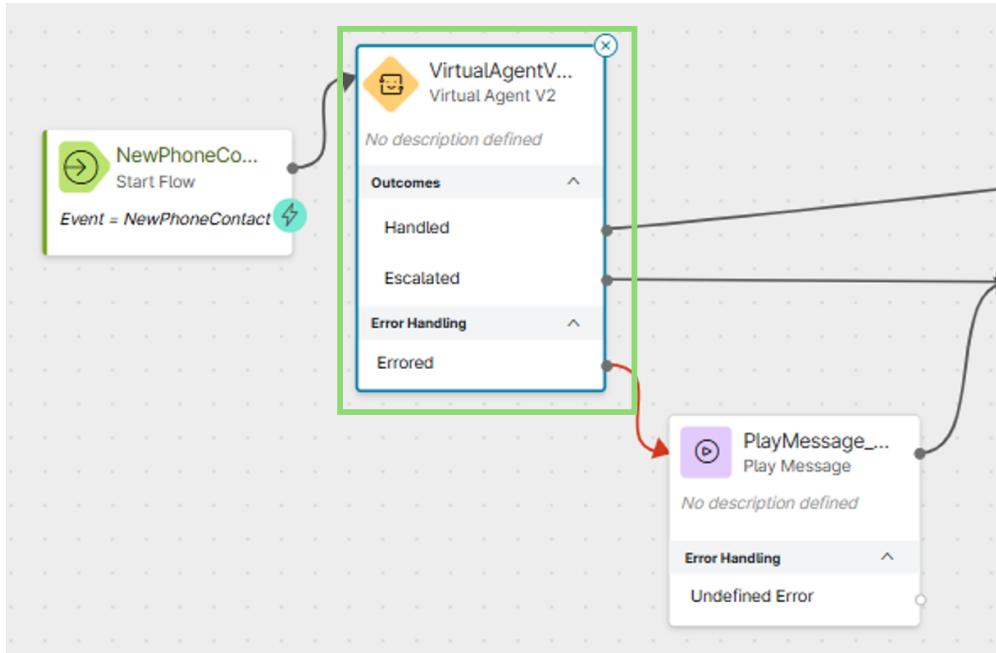
AI agent configuration Published

Add actions to connect your agent with external systems, or link a knowledge base to help it answer questions.

Profile Knowledge Actions Language

Agent name *	URL for agent profile image *
Anuj_WebexOne_Car	https://studio.aiagent-us1.cisco.com/static/assets/img/...
System ID *	AI engine *
Anuj_WebexOne_Car-8pGzpqCi	Webex AI Pro 1.0 Ideal for most contact center use cases with global language support and human-like interactions.
Agent's goal *	Welcome message *
Answer Support Questions	Hi there, how can I help you?
Instructions	
Be polite, as the client is important. Answer the question in no more than 50 words. <small>Thank the user for their business and say goodbye</small>	

- Second you may notice that the Agent may quickly conclude the call without taking an order or completing the transaction.
- This is likely related to the termination delay setting, which may be set for too small a window.
- To fix this, follow these steps:
- In the Flow section of the voice flow you created, select the VirtualAgentV2 node.



- Under Advanced Settings, you'll see the timeout is set to 15 seconds.

State Event**Advanced Settings** Termination Delay 

15

seconds

Speaking Rate 

1.0

Enter a static value or expression that resolves to a number between 0.25 and 4.0. If typing an expression, use this syntax: {{ variable }}

- Increase this timeout to 30 seconds and publish the flow.

 **Note**

Detailed instructions on how to edit and republish flows have not been provided intentionally, as these steps were performed earlier in the lab.

- Retest the flow by calling in again and place a successfull order.

Congratulations! - You have completed this task and the lab!

1.6 Excercise 4 - Explore Webex AI Assistant Feature

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

Control Hub URL	https://admin.webex.com
Username	labuserID@wx1.wbx.ai (<i>where ID is your assigned pod number; this ID will be provided by your proctor</i>)
Password	webexONE1!

1.6.1 Objective

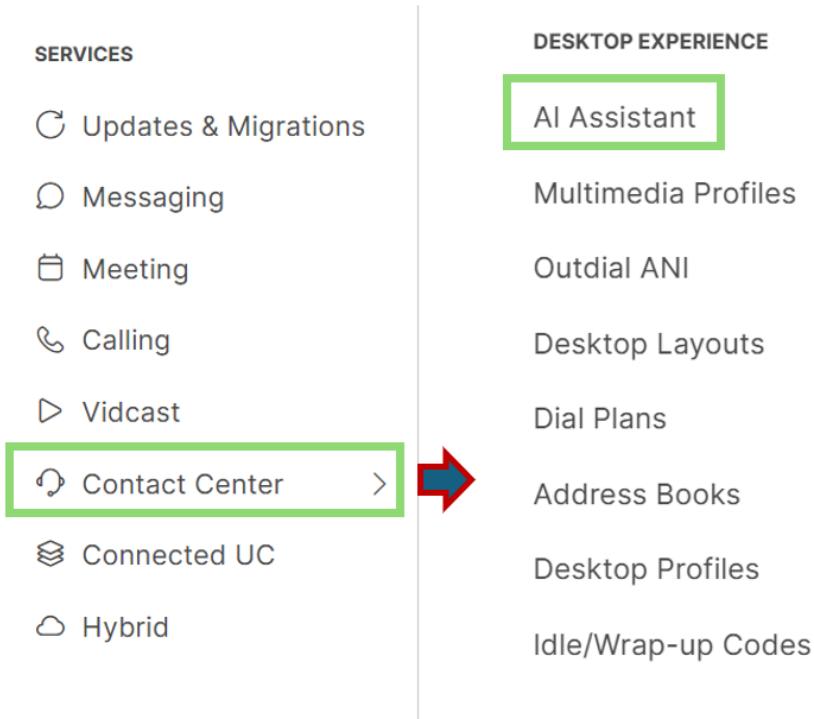
This lab's objective is to provide participants with a basic understanding of the AI Assistant feature in the Webex Contact Center and demonstrate how it enhances the live agent experience.

Building on previous labs that showcased how Webex AI Agents assist customers with inquiries and actionable tasks, this lab will focus on the AI Assistant's role when a customer's call is escalated to a live agent. You will explore, configure, and test the following AI capabilities:

- **AI Agent Transfer Summaries** : See how the AI provides a summary of the conversation when a virtual agent transfers a call to a human agent, giving the agent full context.
- **Call Drop Summaries** : Learn how the AI generates a summary for a new agent if a customer's call is disconnected and they call back, ensuring they don't have to repeat themselves.
- **Real-time Transcriptions** : Experience how this feature provides a live, real-time transcription of the conversation for agents , improving clarity and customer satisfaction.

1.6.2 Section 1: AI Agent Transfer Summary

- To achieve this objective, we must first ensure that the all the summary and transcripts feature is enabled for the tenant.
- To check this, Log in to <https://admin.webex.com> with the provided credentials.
- In Control Hub, under Services, click on Contact Center. Then, under Contact Center navigation pane on the left side, scroll down to DESKTOP EXPERIENCE section and click on Cisco AI Assistant.



- Ensure that the "Call Drop Summary" , "Virtual Agent Transfer Summary" & "Real Time Transcripts" toggle is enabled.

Generated summaries

Streamline agent collaboration and ensure smoother customer experiences with AI-generated call summaries.

[How does this work? ↗](#)

Call Drop Summary

Virtual Agent Transfer Summary

Real-time Transcriptions

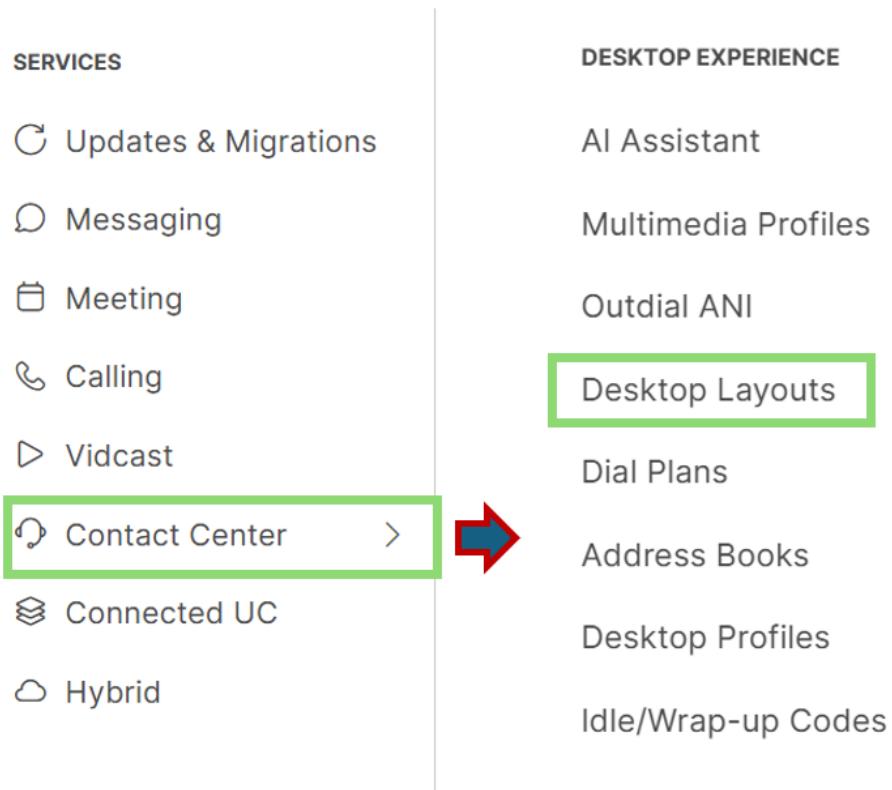
Boost efficiency with accurate real-time transcriptions of customer interactions.

[How does this work? ↗](#)

Real-time transcriptions

For queues that require transcriptions, add media forking activity in Flow Designer. [Manage flows](#)

- The next step is to configure the AI Assistant widget for agent use.
- To accomplish this, the agent's desktop layout must include the "ai-assistant" JSON string in the Agent > Advanced Header section
- You can review this in Control Hub by navigating to "**Desktop Layouts**" within the "**Desktop Experience**" section of the Contact Center services.



- Search and Open the layout **Wx1/Desktop/AI/Assistant** and download the file **Wx1/Desktop/AI/Assistant.json**.

< Desktop Layouts

Wx1/Desktop/AI/Assistant

ID: 2f91ee1a-ab0b-4555-9f86-43042580afe0 • Last Modified: August 18, 2025 15:52 PM

Active

Layout Details

Name * Wx1/Desktop/AI/Assistant

Description Enter a description

Teams WebexOne_Team_1

Type here

1 Teams

JSON File *

Please use the default desktop layout to customize your desktop.

Wx1/Desktop/AI/Assistant.json uploaded

Replace file

- Review the file in a text editor.
- In the **Agent > Advanced Header section**, you should see the verbiage highlighted in this screenshot

```

{
  "agent": {
    "version": "0.1.0",
    "appTitle": "Webex Contact Center",
    "logo": "",
    "stopNavigateOnAcceptTask": false,
    "dragDropEnabled": false,
    "notificationTimer": 8,
    "maximumNotificationCount": 3,
    "browserNotificationTimer": 8,
    "wxmConfigured": false,
    "desktopChatApp": {
      "webexConfigured": false
    },
    "area": {
      "advancedHeader": [
        {
          "comp": "digital-outbound",
          "script": "https://wc.imiengage.io/AIC/engage_aic.js",
          "attributes": {
            "darkmode": "$STORE.app.darkMode",
            "accessToken": "$STORE.auth.accessToken",
            "orgId": "$STORE.agent.orgId",
            "dataCenter": "$STORE.app.datacenter",
            "emailCount": "$STORE.agent.channels.emailCount",
            "socialCount": "$STORE.agent.channels.socialCount"
          }
        },
        {
          "comp": "ai-assistant"
        }
      ]
    }
  }
}

```

- To have a desktop layout reflected for agents, it must be mapped to their teams.
- From User Management section of Control Hub, select Teams and search for your team **WebexOne_Team_num** you want to update.

USER MANAGEMENT

Sites

Skill Definitions

Skill Profiles

Teams

User Profiles

Contact Center Users

- In the Desktop Layout field of the team setting, ensure that the layout **Wx1/Desktop/AI Assistant** is selected and saved.

Team settings

Team type *

Capacity Based
 Agent Based

Skill profile

Select ▼

Select if this team will support skills-based routing with assigned skills in flows. [How does this work?](#) ↗

Multimedia profile

Default_Multimedia_Profile ▼

Desktop layout

Wx1/Desktop_AI_Assistant ▼

Agents

webex1wxccuser+1@gmail.com ×

Type here
▼

1 Agents
✖ Clear All

- Now, log in to the Agent Desktop using the provided credentials.
- URL: <https://desktop.wxcc-us1.cisco.com/>
- Username: Contact the lab proctor if information is unavailable.
- Password: Contact the lab proctor if information is unavailable.
- Please select **Desktop** as the telephony option and set the Team as **WebexOne_Team_num**

- 53/78 -

Copyright © 2025 Cisco

Station Credentials

Select your telephony option 

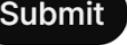
Dial Number Extension Desktop

Desktop allows to receive inbound calls and make outdial calls through the internet.

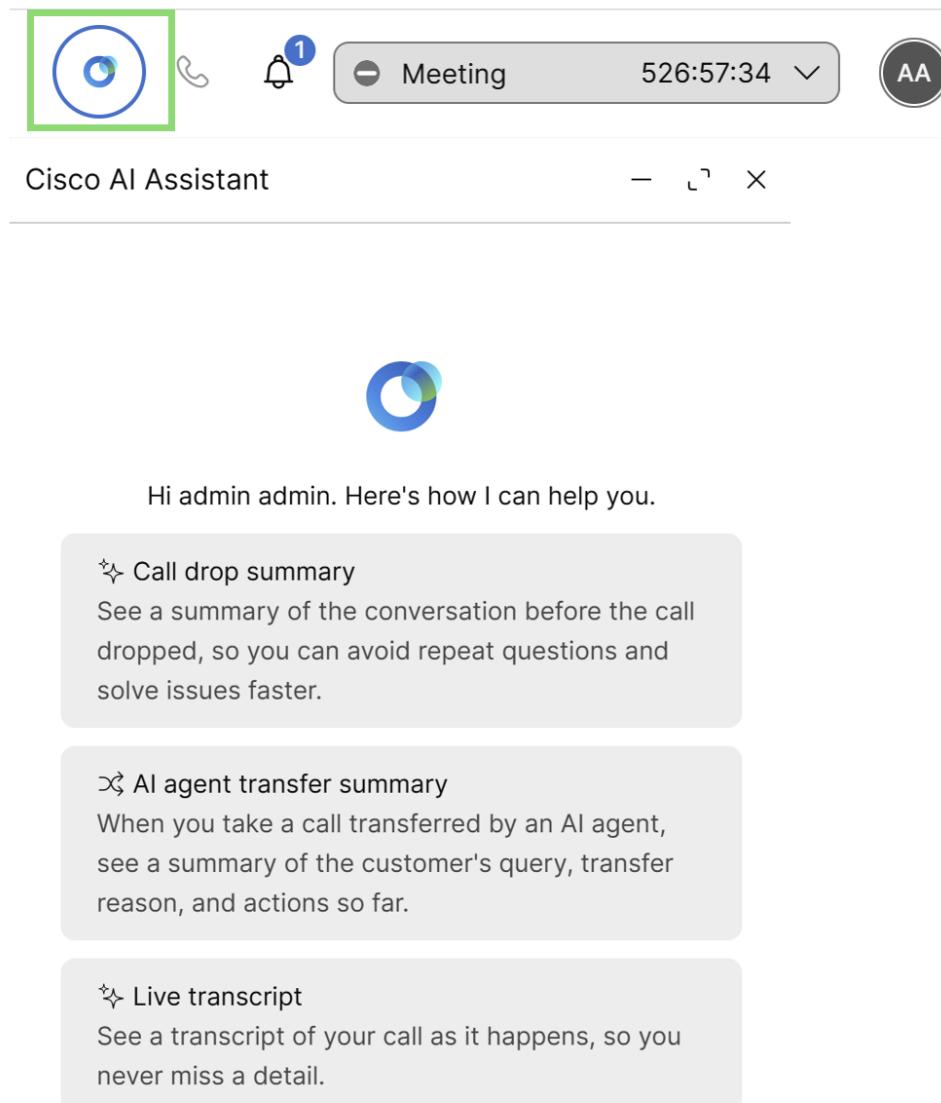
 When working outside the corporate location, make sure to update your current address in the Emergency service. [See help section.](#)

Team  

Remember My Credentials

- This will load the layout as configured above.
- You will notice the AI Assistant and its capabilities loaded.



- Ensure the agent is set to the "**Available**" state.
- Using your cell phone, call the same number as in previous exercise.
- After speaking briefly with the AI agent, you can escalate the call to a live agent by saying, **I want to speak to an agent**.
- Switch the agent to '**Available**' state on the Webex Contact Center Agent Desktop and accept the call.
- Observe the AI Assistant summary of your conversation with the agent and all the interactions with the AI agent in the IVR transcripts section.

The screenshot shows a Cisco contact center interface. At the top, there are several icons: a speech bubble, a blue circle with a phone, a handset, a bell with a red '2', and a checkmark with the word 'Engaged'. To the right of 'Engaged' is a dropdown arrow and a yellow circular button with 'UO'. Below these are two buttons: 'Hold' and 'Consult'. A large call summary box is displayed, containing the text: 'AI agent transfer summary is ready. See a summary of the customer's query to make the transition smoother.' A green-bordered button labeled 'View summary' is at the bottom of this box. On the left, the text 'DNIS' is visible. At the bottom, there is an 'IVR Transcript' section with a history of messages between a 'Virtual Agent' and a 'Customer'. To the right of the transcript is a 'Cisco AI Assistant' window with a summary of the AI agent transfer.

IVR Transcript

02:13

Keypad Mute Hold Consult Transfer Pause Recording End

Phone Number DNIS

Queue AIQueue1

IVR Transcript

Hmm., Great, Washington! Let's find the perfect car for you. Could you tell me what type of car maybe a sedan? And if you have any preferences for color or engine type, let me know!

Virtual Agent 03:57 PM
Really?

Customer 03:57 PM
Fast car, right?

Virtual Agent 03:57 PM
Really?, For a fast car, you might be interested in the Swiftura or SkyLux models. Both are sport The Swiftura comes in Midnight Blue, Pearl White, and Crimson Red, with hybrid and electric en

Cisco AI Assistant

Here's the AI agent transfer summary

The reason for the call was to order a new car.
The call was handed off when the caller requested to talk to an agent.

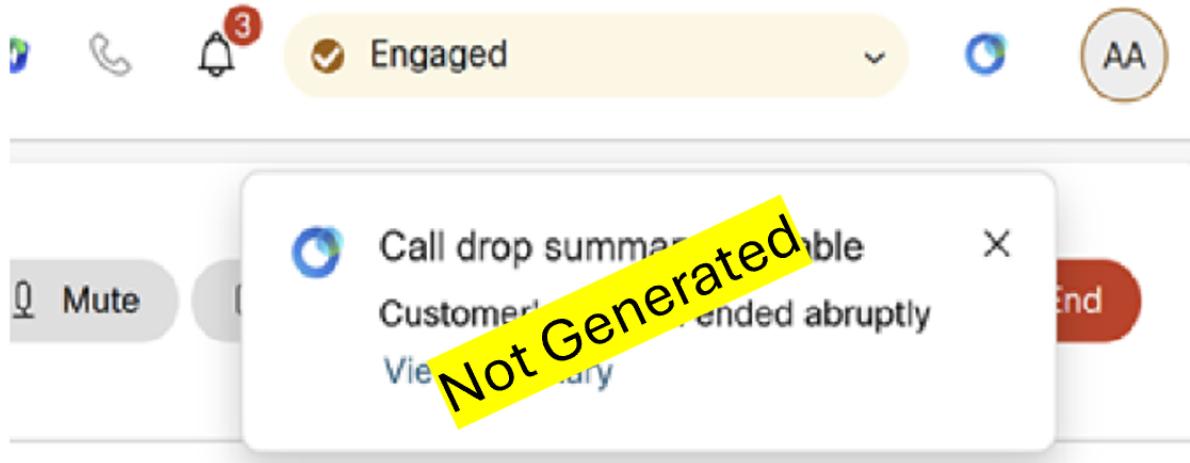
- The caller expressed interest in a fast car.
- The bot agent recommended the Swiftura and SkyLux models.
- The caller chose the Swiftura in Crimson Red.

03:59 PM

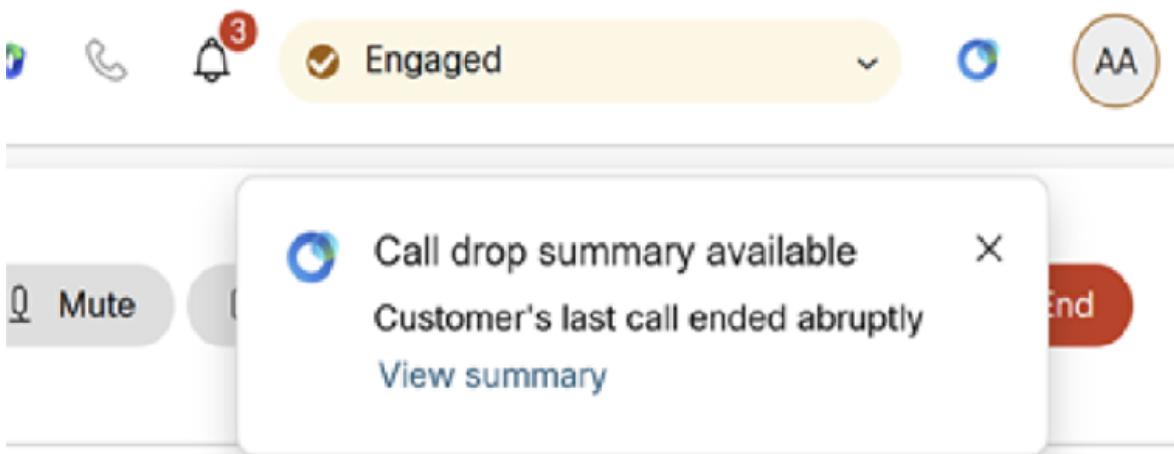
- Have a brief conversation with the agent, ensure it's not more than **10 or 15** seconds long, and end the call from the cell phone.

1.6.3 Section 2: Call Drop Summary

- Now, if another call is made that is escalated and presented to an agent, you might expect a call drop summary to appear when the agent accepts the call.
- Let's make a call and observe the behavior.
- You will notice that the summary is not generated when the call is presented to the agent and they answer it.



- This time, continue the conversation with the agent for more than **40 seconds**, then end the call from the caller's end.
- Now, when you call back and the agent accepts the call, the call drop summary will be generated.



- The reason this works is that the call has met the following criteria:
- Recording for these calls is enabled.
- The callback to the contact center is within a 24-hour window.
- The callback came from the same phone number (ANI) as the original call.
- The call was longer than 30 seconds.
- More details can be reviewed in this section Frequently Asked Questions (FAQs)

1.6.4 Section 3: Real Time Transcription

- Real-time transcriptions feature allows agents to access real-time conversation with the caller on their Agent Desktop, enabling them to follow conversations more accurately and respond effectively.
- While completing the "Call Drop Summary" exercise, you may have noticed that transcripts were not visible for the calls, even though the option was enabled in Control Hub along with the "Call Drop Summary" feature.

Generated summaries

Streamline agent collaboration and ensure smoother customer experiences with AI-generated call summaries.

[How does this work? ↗](#)

Call Drop Summary

Virtual Agent Transfer Summary

**Real-time Transcriptions**

Boost efficiency with accurate real-time transcriptions of customer interactions.

[How does this work? ↗](#)

Real-time transcriptions

For queues that require transcriptions, add media forking activity in Flow Designer. [Manage flows](#)

- This is because, in addition to the tenant-level settings, a "**Media Stream**" activity must be triggered directly after the agent answers the call.
- This can be achieved in a flow in one of two ways:
 - Enable media stream for all queues in the flow.
 - Enable media stream for specific queues in the flow.
- For this exercise, we will proceed by enabling media stream for all queues in the flow as we have only one queue.
- In Control Hub go to the Flows menu on the left-hand side.

CUSTOMER EXPERIENCE

Channels

Queues

Business Hours

Audio Files

Flows

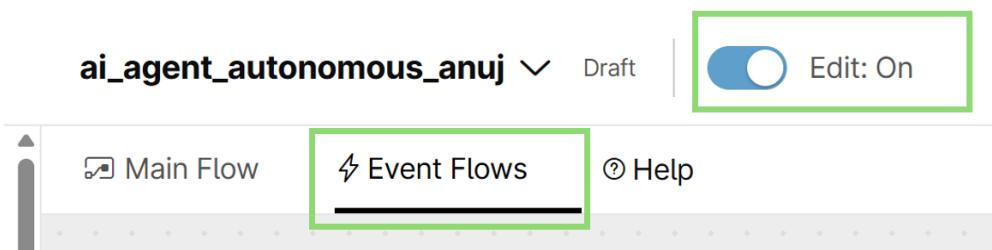
AI Agents

Call Recording Schedules

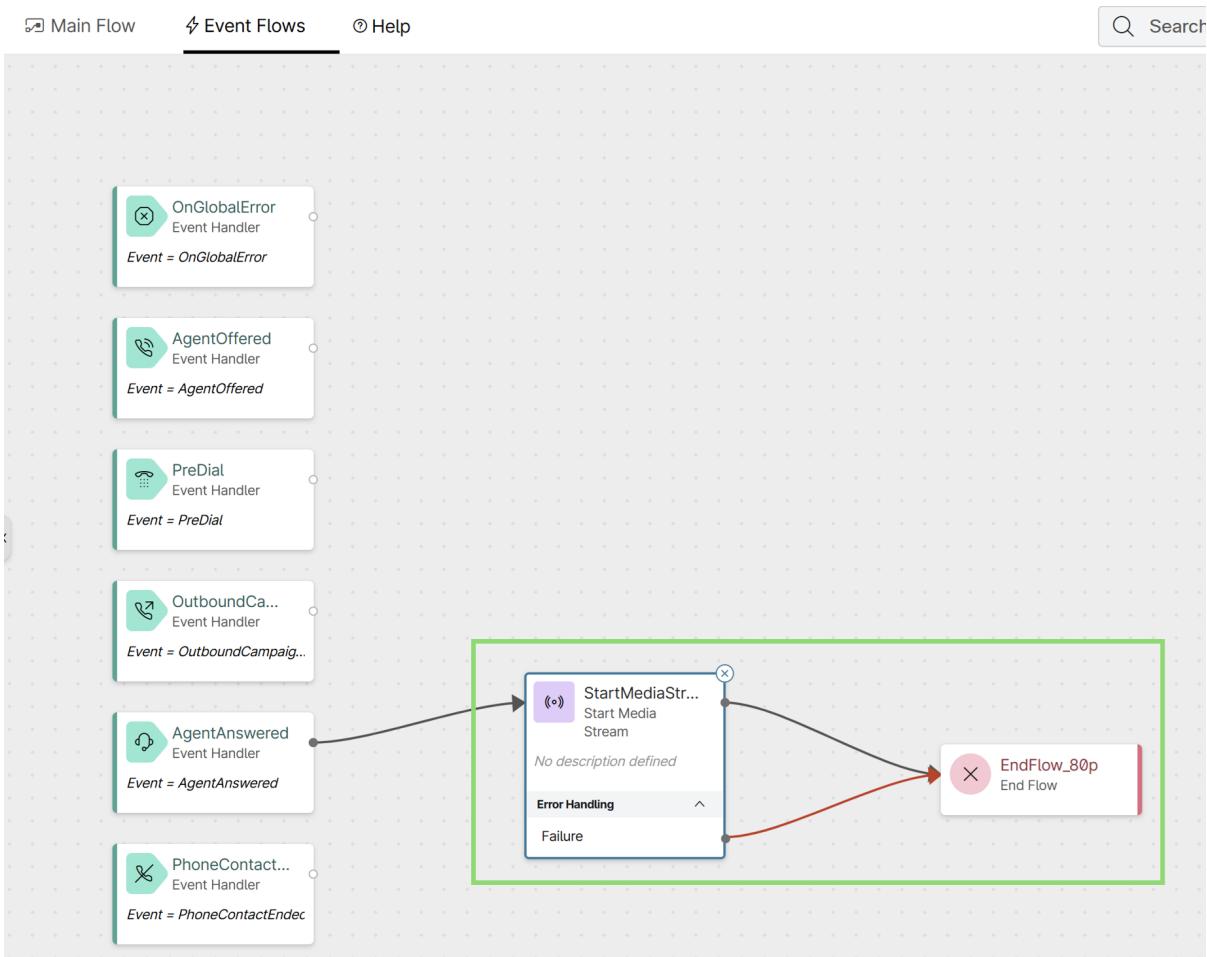
Functions

Surveys

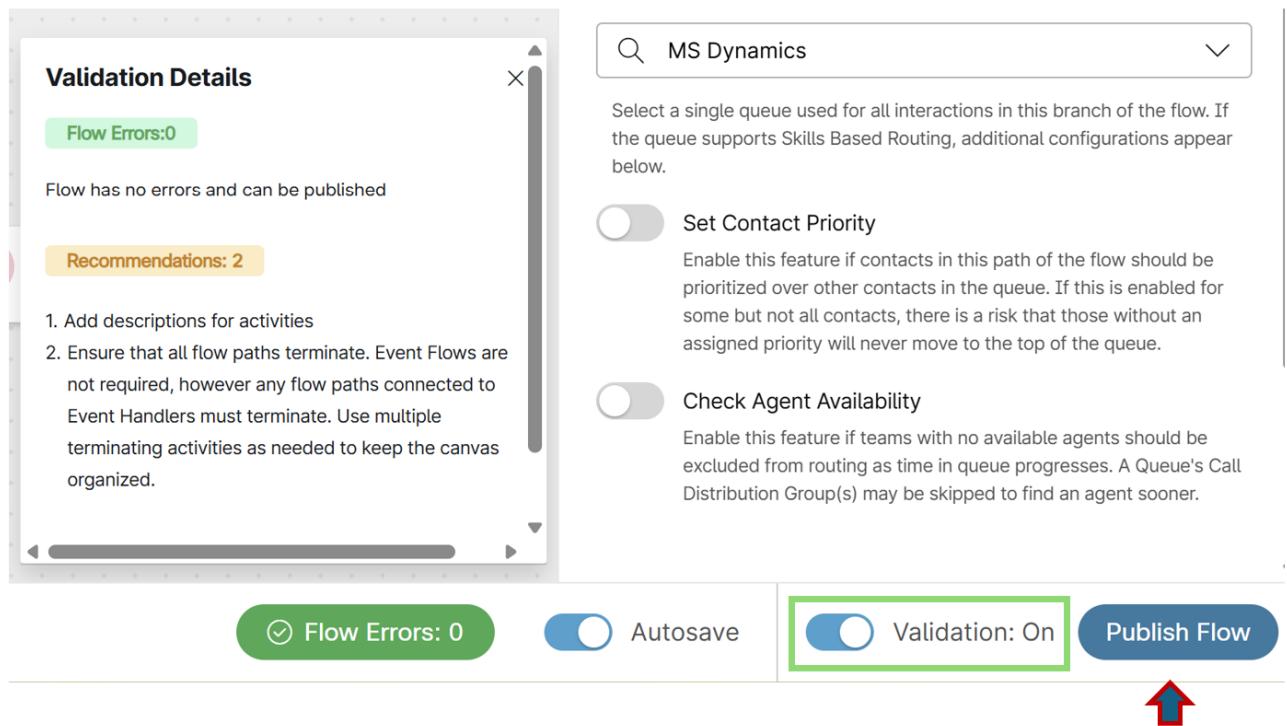
- Search for and select the voice flow you created for the AI Agent exercise.
- On the event flow canvas, click the Edit option.



- Add a Start Media Stream activity node directly after the AgentAnswered event. Make sure the activity is connected to the End Flow Node.



- Validate and Publish the flow. This will enable real-time transcription for every call routed through this flow.



- Now, let's test the transcription feature
- Ensure your agent is logged in and set to available.
- Make a call and after the interaction with the AI agent, escalate the call to your agent.
- Accept the call and observe the conversation being converted into real-time transcripts in the Transcripts gadget.

≡ IVR Transcript ⏲ Contact History **≡ Live Transcript**

Live Transcript

You started a chat with Customer. 09/18/2025 11:51 AM

c Customer 11:51 AM
Hi.

>You 11:51 AM
How I, how are you?

c Customer 11:51 AM
How are you how are you doing? How are you doing? I'm doing great. Great. Good. Good.

>You 11:51 AM
I'm doing great.

- If your business requires enabling real-time streaming for specific queues only, you can follow the instructions in the provided document at [Enabling media streaming for specific queues](#)

1.7 Results

Congratulations, In this lab, you learned how the AI Assistant enhances the live agent experience. By providing AI Agent Transfer Summaries and Call Drop Summaries, the assistant gives agents instant context. Paired with Real-time Transcriptions, these features ensure agents are always well-informed, leading to more efficient and satisfying customer interactions.

1.8 Excercise 5 - (Bonus) Car Order Actions with Webex AI Agent

Please use the following credentials to connect to Control Hub and configure Webex Contact Center:

Control Hub URL	https://admin.webex.com
Username	labuserID@wx1.wbx.ai (where ID is your assigned pod number; this ID will be provided by your proctor)
Password	webexONE1!

1.8.1 Steps

To make the AI Agent more capable, additional bot actions are needed. The configuration process is very similar to the one completed above for order creation. For simplicity, Webex Connect flows for these next actions are already prebuilt and will not require manual configuration.

Car order check action

- In the AI Agent Studio, navigate to your Autonomous Webex AI Agent.
- Select '**Actions**' and click '**New Action**'.
- Provide '**Action Name**' (use *carordercheck*).
- Provide the following '**Action description**':

Check an existing car order and provide details about the order - model, color, engine type, and orderid.

- Under '**Action scope**', select '**Slot filling and fulfillment**'.

General information

Action name *

Action description * ⓘ

Check existing car order and provide details about the order - type, color, engine type, id, orderid.

Action scope *

Slot filling and fulfillment

- Select the prebuilt service *CiscoLive25AIAgentFlows* under '**Select service**' and the *carordercheck* flow under '**Select a flow**'.

Webex Connect Flow Builder Fulfillment

Determine how to handle customer requests or resolve issues.

[Learn more about Webex Connect Flow Builder for fulfillment](#)

Select service *

CiscoLive25AIAgentFlows

Select a flow *

carordercheck

- Click 'New input entity'.

Entity name	Entity type	Entity description	Entity examples	Required
orderid	String	orderid provided by the customer	5345321	Yes

Slot filling

Add input entities in a table or define input parameters in JSON for slot filling to ensure the AI agent gathers all necessary information.

Input entities [Use JSON instead](#)

+ New input entity

Entity Name	Type	Value	Description	Example	Required	Controls
orderid	string	-	orderid that customer provides	5345321	Yes	

- Click 'Add'.

Car order modify action

- In the AI Agent Studio, navigate to your Autonomous Webex AI Agent.
- Select 'Actions' and click 'New Action'.
- Provide 'Action Name' (use *carordermodify*).
- Provide the following 'Action description':

Collect the orderid and modify the order as per the customer's request.

- Under 'Action scope', select 'Slot filling and fulfillment'.

General information

Action name *

carordermodify X

Action description * ⓘ

Collect order number and modify the order as per customer request.

Action scope *

Slot filling and fulfillment ▼

- Select the prebuilt service *CiscoLive25AIAGentFlows* under '**Select service**' and the *carordermodify* flow under '**Select a flow**'.

Webex Connect Flow Builder Fulfillment

Determine how to handle customer requests or resolve issues.

[Learn more about Webex Connect Flow Builder for fulfillment](#)

Select service *

CiscoLive25AIAgentFlows

Select a flow *

carordermodify

- Add five entities for the Webex AI Agent to pass to the Webex Connect flow.

- Click 'New input entity'.

Entity name	Entity type	Entity description	Entity examples	Required
orderid	String	Order Id of the existing order	5345321	Yes
model	String	Car model that the customer chose	NovaTerra	No
color	String	Car color that the customer chose	Ruby Red	No
engine	String	Engine type that the customer chose	gas, electric	No
customer	String	Customer's name	Jack, Nick	No

- Click 'Add'.

Slot filling

Add input entities in a table or define input parameters in JSON for slot filling to ensure the AI agent gathers all necessary information.

Input entities

[Use JSON instead](#)

Entity Name	Type	Value	Description	Example	Required	Controls
color	string	-	Car color that the customer chose	Ruby Red	No	
customer	string	-	Customer's name	Jack, Nick	No	
engine	string	-	Engine type that the customer chose	gas, electric	No	
model	string	-	Car model that the customer chose	NovaTerra	No	
orderid	string	-	Order Id of the existing order	5345321	Yes	

Car order delete action

- In the AI Agent Studio, navigate to your Autonomous Webex AI Agent.
- Select '**Actions**' and click '**New Action**'.
- Provide '**Action Name**' (use *carorderdelete*).
- Provide the following '**Action description**':

Delete the order when requested.

- Under '**Action scope**', select '**Slot filling and fulfillment**'.

The screenshot shows the 'General information' section of the 'New Action' configuration. It includes fields for 'Action name' (set to 'carorderdelete'), 'Action description' (set to 'Delete the order when requested.'), and 'Action scope' (set to 'Slot filling and fulfillment'). The 'Action description' field is highlighted with a blue border.

General information

Action name *

carorderdelete

Action description * ⓘ

Delete the order when requested.

Action scope *

Slot filling and fulfillment

- Select the prebuilt service *CiscoLive25AIAGentFlows* under '**Select service**' and the *carorderdelete* flow under '**Select a flow**'.

Webex Connect Flow Builder Fulfillment

Determine how to handle customer requests or resolve issues.

[Learn more about Webex Connect Flow Builder for fulfillment](#)

Select service *

CiscoLive25AIAgentFlows

Select a flow *

carorderdelete

- Click 'New input entity'.

Entity name	Entity type	Entity description	Entity examples	Required
orderid	String	Orderid provided by the customer	5345321	Yes

- Click 'Add'.

Slot filling

Add input entities in a table or define input parameters in JSON for slot filling to ensure the AI agent gathers all necessary information.

Input entities [Use JSON instead](#)

Entity Name	Type	Value	Description	Example	Required	Controls
orderid	string	-	Orderid provided by the customer	5345321	Yes	

- Save the Webex AI Agent configuration by clicking '**Save Changes**', and publish by clicking '**Publish**' (provide comments for version tracking purposes).

AI agent configuration Ready to preview

Add actions to connect your agent with external systems, or link a knowledge base to help it answer questions.

Profile **Knowledge** **Actions** **Language**

Publish

Publish and track changes

Add a brief comment to explain this update. It will appear in the version history.

Comment *

v1

Cancel **Publish**

- Congratulations! You have completed this task.

1.9 Conclusion

Congratulations on completing the Autonomous AI agent lab! You've successfully learned how an AI Assistant can significantly enhance the live agent experience. By providing an AI Agent Transfer Summary and a Call Drop Summary, the assistant gives agents instant context, allowing them to quickly understand the customer's needs. When combined with real-time transcriptions, these features ensure agents are always well-informed, leading to more efficient and satisfying customer interactions.

You have also successfully completed the exercise of enhancing the AI bot to fulfill actions! This is a huge step in building a more capable and autonomous agent.

2. How to use this guide

2.1 Setting up and using the tool

2.1.1 Initial Setup

This lab guide template uses MKDocs to take your markdown documentation and transform it into an interactive web based lab guide. You will need to install some software on your PC or Mac in order to take full advantage of the tool which will allow you to view your changes as you save them and ensure that your formatting is exactly how you want it.

Prerequisite Software to Install

1. Python
 2. Visual Studio Code
 3. Git
-

Setting up your lab guide on your computer

CLONE YOUR REPOSITORY WITH GIT

Open Visual Studio Code



Click the Source Control button in the left menu

Click Clone Repository

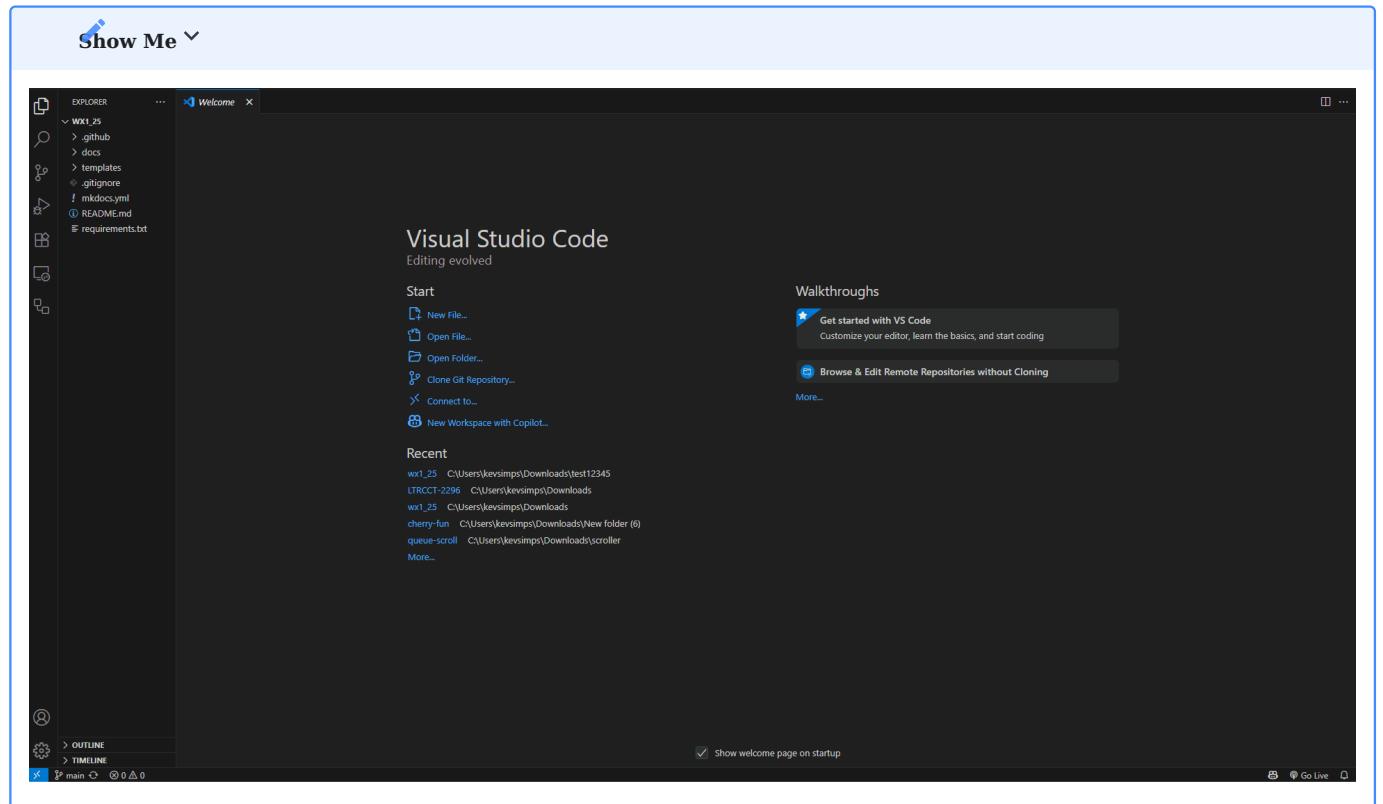
Enter the repository source: {{config.extra.repo}} A small blue clipboard icon with a white document symbol.

Select or create a new folder to clone the repository into.

BUILD THE ENVIRONMENT

When prompted to open the cloned repository, select open.

Drag open the terminal at the bottom of the Visual Studio Code window



If you are on a PC

If you are on a Mac

In your terminal enter the following commands:

`python -m venv venv`

`Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser`

`venv\Scripts\activate.ps1`

`pip install -r requirements.txt`

`mkdocs serve`

Open a browser to `http://127.0.0.1:8000`

In your terminal enter the following commands:

`python -m venv venv`

`source venv/bin/activate`

`pip install -r requirements.txt`

`mkdocs serve`

Open a browser to `http://127.0.0.1:8000`

2.1.2 Using the tool after the initial setup

After the initial setup you will only need to take the following steps:

Open Visual Studio Code



Click the Source Control button in the left menu

Click Open Folder and navigate to the folder where you cloned the repository

If you are on a PC

If you are on a Mac

In your terminal enter the following commands:

venv\Scripts\activate.ps1

mkdocs serve

Open a browser to <http://127.0.0.1:8000>

In your terminal enter the following commands:

source venv/bin/activate

mkdocs serve

Open a browser to <http://127.0.0.1:8000>

2.1.3 Publishing Your Changes

As you progress with the creation of your lab guide, you should sync your changes periodically to GitHub.

Syncing your changes to GitHub

Make sure that you have saved all of the files you want to sync up to the repository



Click the Source Control button in the left menu

Click the plus sign next to each file you want to sync as you hover over the file names or alternatively, click the plus sign next to changes

Enter a note about the changes you are making in the message text box

Click the down chevron on the commit button to reveal more options

Select Commit & Sync

AFTER YOU SYNC YOUR CHANGES, IT WILL TAKE A FEW MINUTES FOR GITHUB ACTIONS TO BUILD THE WEBSITE SO THAT YOU CAN VIEW ALL OF THE CHANGES YOU MADE.

2.2 Styling and Advanced Features

2.2.1 Markdown and Other References

MK DOCS FEATURE REFERENCE:

<https://squidfunk.github.io/mkdocs-material/reference/>

MARKDOWN CHEAT SHEETS:

<https://www.markdownguide.org/cheat-sheet/>

<https://github.com/lifeparticle/Markdown-Cheatsheet>

2.2.2 Custom Features

Adding a Copy Button Without a Code Block

This will copy the following text: <copy>Text to copy</copy>

This will copy the following text: Text to copy 

Adding user specific variables into your lab guide

There may be times in which you want to embed some attendee specific information into the instructions of your lab guide, like credentials or phone numbers, which will be used on multiple pages of your lab guide. You can gather the information via a form or you can pass a pre-encoded JSON string as a URL parameter to populate the variable values into the browser's session storage. Then you can use a simple HTML tag with a special class name to update the values in the lab guide. This feature can be combined with the copy button method above.

FORM METHOD

Show me the code

```
<form id="info">

<label for="Admin">Admin Login:</label>
<input type="text" id="Admin" name="Admin"><br>

<label for="PW">Admin Password:</label>
<input type="text" id="PW" name="PW"><br>

<label for="EP">Inbound Channel Name:</label>
<input type="text" id="EP" name="EP"><br>

<label for="DN">Inbound Channel Phone Number:</label>
<input type="text" id="DN" name="DN"><br>

<label for="Queue">Queue 1 Name:</label>
<input type="text" id="Queue" name="Queue"><br>

<label for="Queue2">Queue 2 Name:</label>
<input type="text" id="Queue2" name="Queue2"><br>

<label for="Team">Team 1 Name:</label>
<input type="text" id="Team" name="Team"><br>

<label for="Team2">Team 2 Name:</label>
<input type="text" id="Team2" name="Team2"><br>
<br>
<button onclick="setValues()">Update Lab Guide</button>
</form>
```

Example Input Form**Admin Login:****Admin Password:****Inbound Channel Name:****Inbound Channel Phone Number:****Queue 1 Name:****Queue 2 Name:****Team 1 Name:****Team 2 Name:****Update Lab Guide**Login: Provided by proctor Password: Provided by proctor Assigned Inbound Channel Name: Provided by proctor Assigned Inbound Channel Number: Provided by proctor Assigned Queue Name 1: Provided by proctor Assigned Queue Name 2: Provided by proctor Assigned Team name 1: Provided by proctor Assigned Team name 2: Provided by proctor **URL METHOD**

If you have a lot of attendee variables in your lab, you may choose to precompile and encode them so that you can simply provide a URL link which will load all of their required information.

To see this in action, add this string at the end of the URL for any page on this site: ?

eyJZG1pbil6ImFkbWluQHh5ei5iaXoiLCJVyI6InNVcGVyU2VjcmV0MTIzISIsIkVQIjoiRVAxIwiRE4iOiIrMTkxMDU1NTEyMTUyIiwiUXVldWUiOjRdWV1ZTEiLCJRdWV1ZTiiOjRdWV1ZTiiLCJUZWftIjoiVGVhbTEiLCJUZWftMii6IlRlYW0yIn0= 

2.2.3 Custom Admonitions

In addition to the **stock admonitions** which are available out of the box we have added some additional branded admonitions which were created by Bobby McGonigle.



Blank

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Blank!')
```



Vidcast

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Vidcast!')
```



Download

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Download!')
```



Ce-deploy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Ce-Deploy!')
```



Webex

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Webex!')
```



Gif

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('How should I pronounce Gif?')
```

 **Important**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Important!')
```

 **Challenge**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Challenge!')
```

 **Tool**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Tool!')
```

 **Curious**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla et euismod nulla. Curabitur feugiat, tortor non consequat finibus, justo purus auctor massa, nec semper lorem quam in massa.

```
console.log('Hello Curious!')
```