
AGENT BUILDERS

Build your first AI agents with Copilot Studio

A step-by-step guide for your first go at Copilot Studio.

Version 2 · Workshop edition

Before you start

Read this page first.

What you'll build today

1. **Agent 1 — a knowledge grounded agent** that answers questions from an HR policy document.
2. **Agent 2 — a workflow agent** that saves a PDF to SharePoint and posts a message in Teams.
3. **Agent 3 — a computer-use agent** that browses a website on your behalf and retrieves key information.

Housekeeping

1. Sign in with the workshop account on your paper slip. Do not use your own – or your work - Microsoft account.
2. Work in one private browser window and keep it open. Close it and you'll be signed out of everything.
3. Put your name on every agent and flow you make. We're all in one tenant, so it gets confusing fast otherwise.
4. Things take a minute or two to provision. Leave them to it, don't refresh or click twice.
5. Please don't change any passwords, permissions, connections or settings the guide doesn't mention.

A grey box contains text to copy. Copy everything inside the box, exactly as it is written.

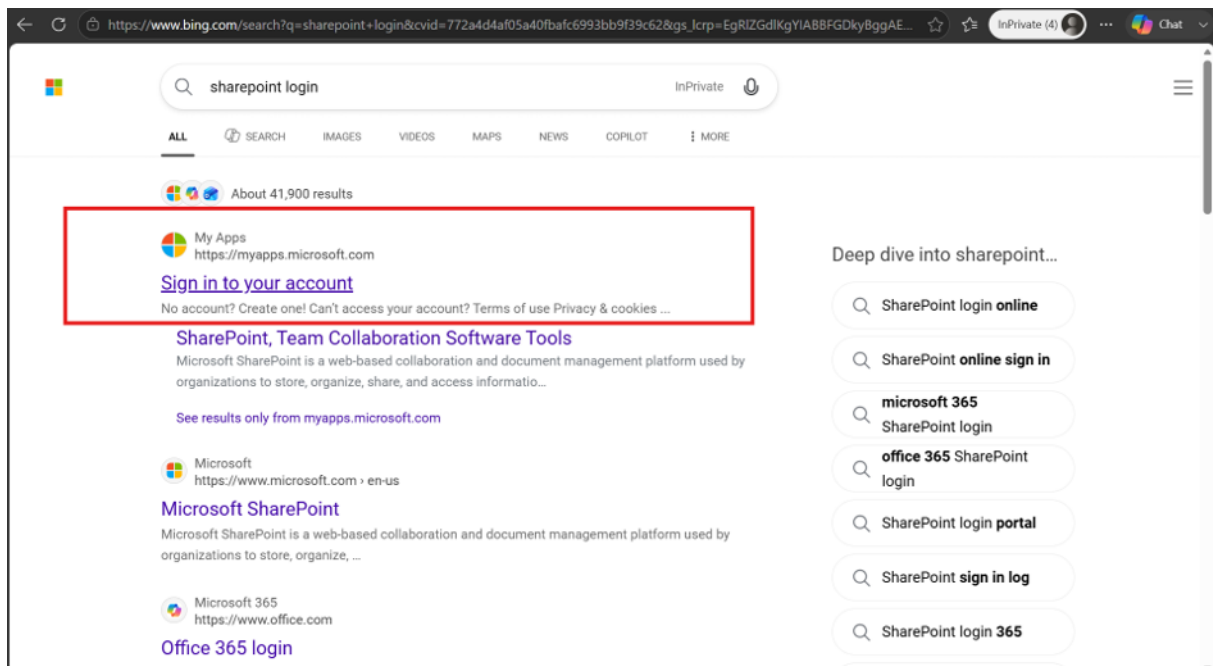
Setup: Sign in and download the workshop files

Everything happens in one private browser window. Set that up now.

1. Open a private window in your browser.

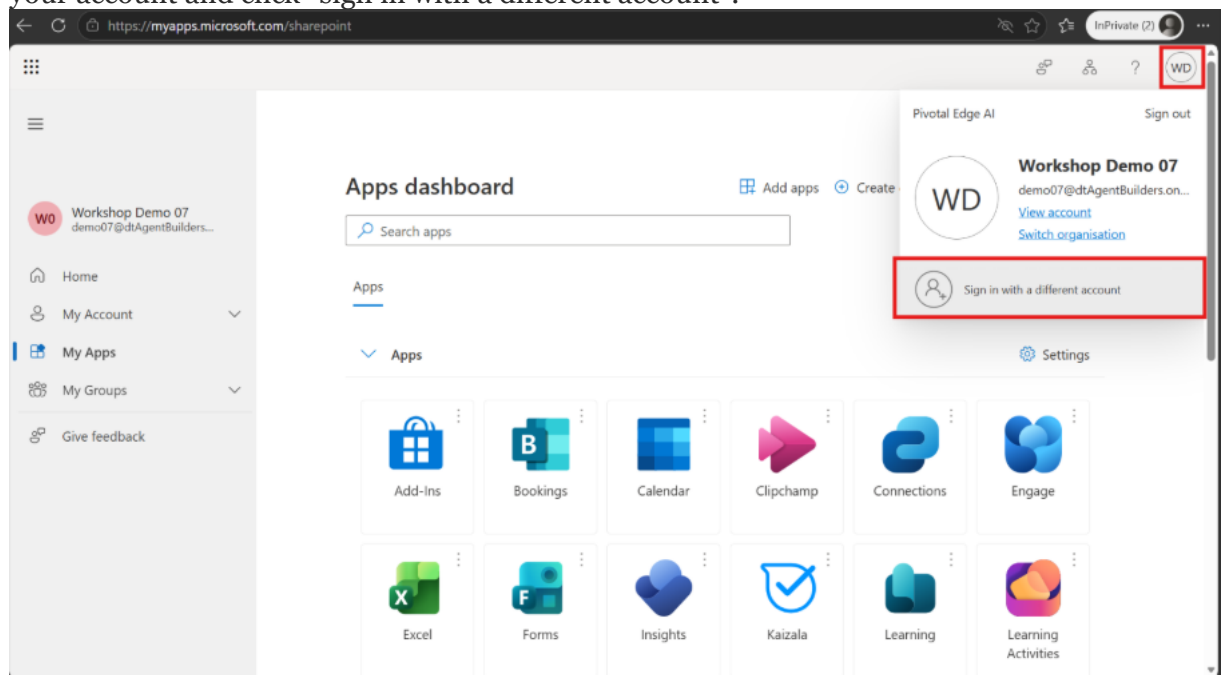
⚠ Everything from here on happens in this one private window. Don't open these pages in your normal browser window.

2. Click into the address bar at the top, type "**SharePoint login**", and press **Enter**.
3. On the results page, click the **Sign in to your account** link, shown in the red box below. It sits under **myapps.microsoft.com**.

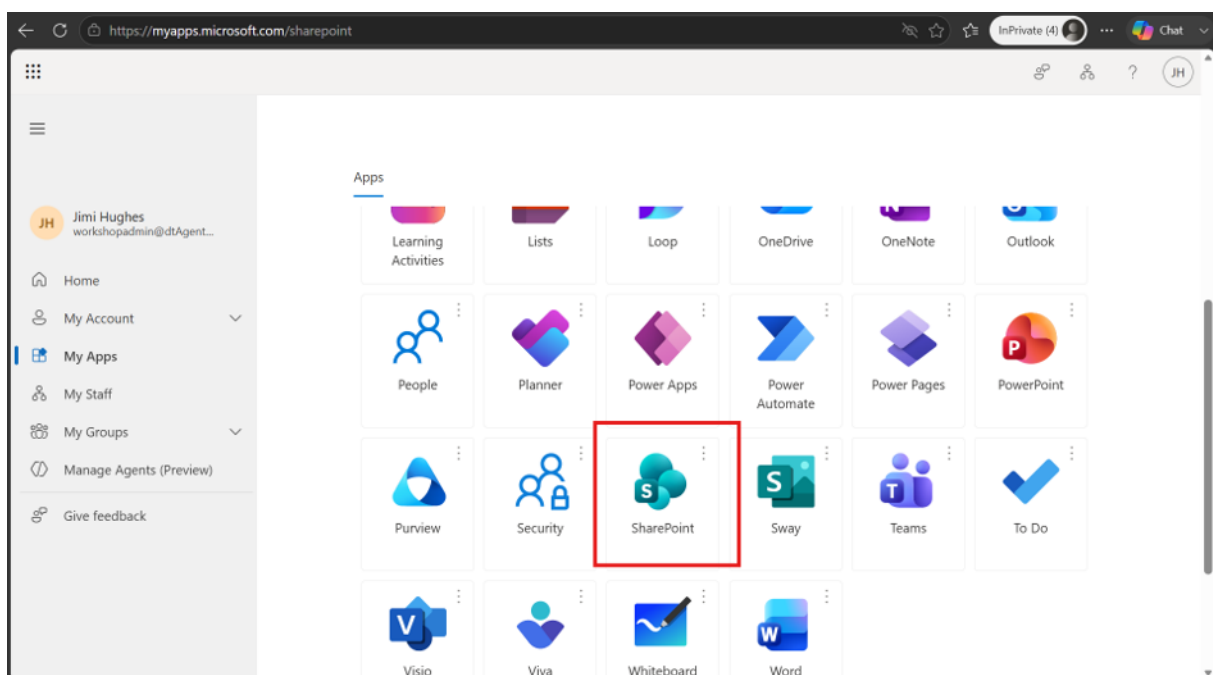


Click the "Sign in to your account" result, not the SharePoint one below it.

4. If you have been automatically signed into one of your own accounts, then please select your account and click “sign in with a different account”.



5. Enter the email address and password from your paper slip. If you are asked to stay signed in, click **Yes**.
6. You're now on the My Apps page. Scroll down and click the **SharePoint** tile, shown in the red box below.

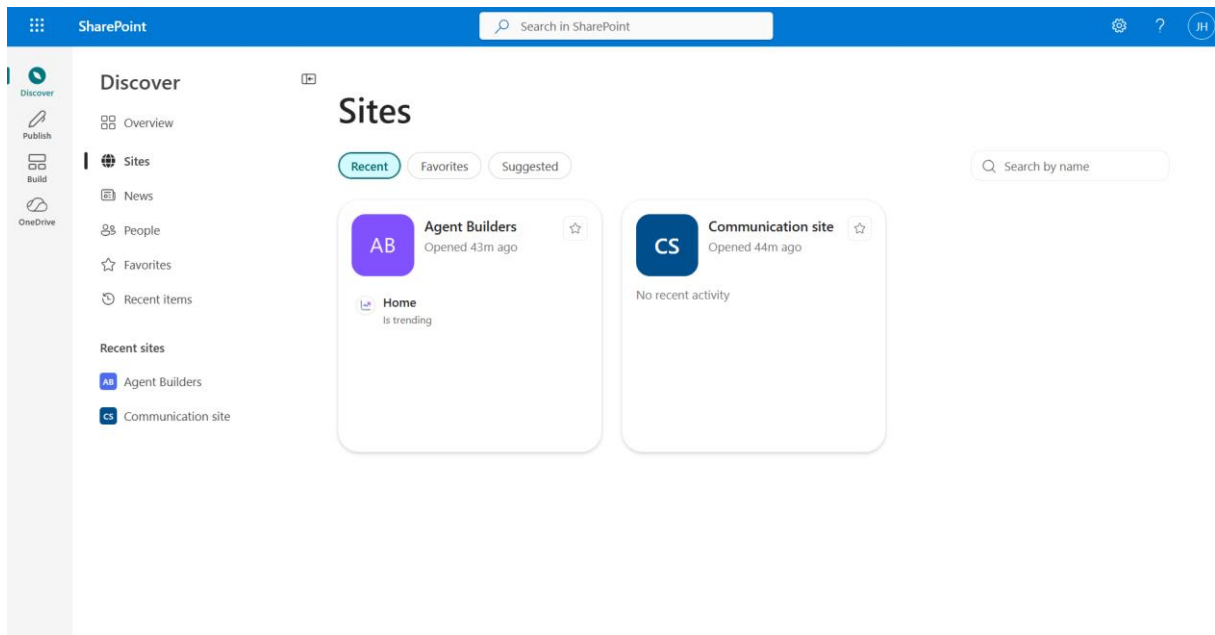


The SharePoint tile on the My Apps page.

⚠ Leave this My Apps tab open. You'll come back to it in Agent 2 to open Teams.

7. In the left menu click **Sites**. Then click the **Agent Builders** card in the middle of the page.

8. If you can't see the agent builders site in SharePoint, then use this link below. Make sure to right click, copy the link and then paste it into your private browser window – otherwise it might open in your normal browser and automatically sign into your personal account. [Agent Builders - Home](#)



Sites in the left menu, then the Agent Builders card.

9. You're now on the Agent Builders site. You'll come back here in Agent 2 to check the file your agent uploads.
10. Leave this SharePoint tab open.
11. Open another new tab and go to the workshop downloads page:

Copy this:

`https://agentbuilders.pro/[DOWNLOAD PAGE - TBC]`

12. Download both files from that page:

The two files you need:

Sample HR policy.docx
Sample PDF.pdf

13. Check where they saved. On most machines this is your **Downloads** folder.

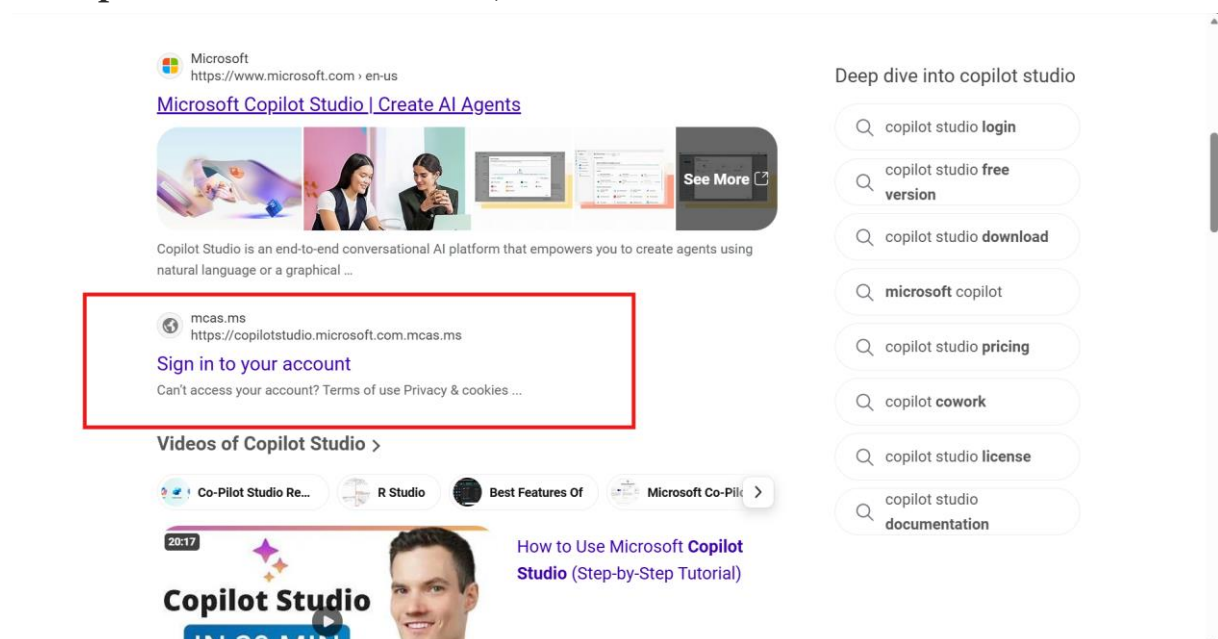
✓ **CHECK:** You have two files in your Downloads folder: the HR policy and a PDF. You'll need the HR policy for Agent 1 and the PDF for Agent 2.

Agent 1: Build a knowledge agent

This agent answers questions from a document you give it. You'll upload an HR policy, then check it answers from the policy and admits when it doesn't know, rather than making something up. This is a simple demonstration of knowledge grounded agents in Copilot Studio.

Open Copilot Studio

1. Open a new tab in the same private window. Type **Copilot Studio** in the address bar and press **Enter**.
2. Scroll down the results and click **Sign in to your account** under **copilotstudio.microsoft.com**, shown in the red box below.

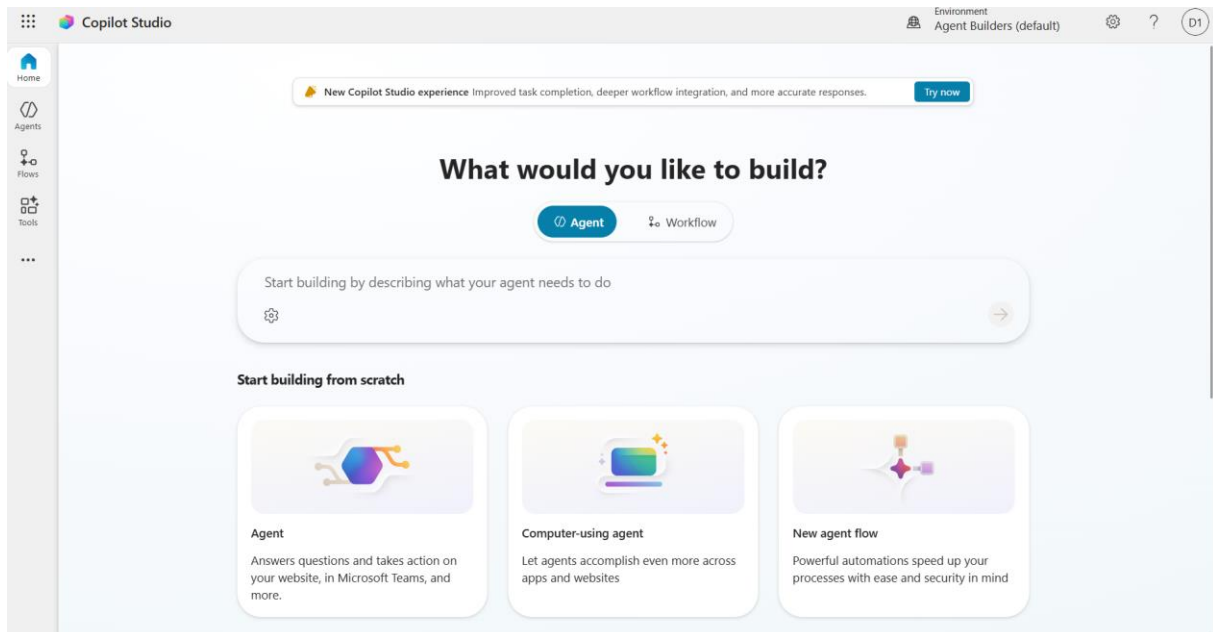


Click the copilotstudio.microsoft.com sign-in link.

3. Sign in with the same workshop account from your paper slip.
4. If a blue banner appears offering the **New Copilot Studio experience**, click the **X** to close it.
5. If you land on the new experience by default, then make sure to switch back to the old interface or the screenshots in this guide won't make sense.

⚠ Don't click "Try now" on that banner. The new experience is not used in this workshop and the screens won't match this guide.

6. Should look like this below



Create the agent

1. In the left menu click **Agents**.
2. Click **+ New agent**.
3. Click **Blank agent**.
4. Wait a minute or two while it provisions. Don't refresh the page.
5. In the **Name** box, type this and replace [your name] with your actual name:

Copy this:

Knowledge Agent - [your name]

6. In the **Description** box, type:

Copy this:

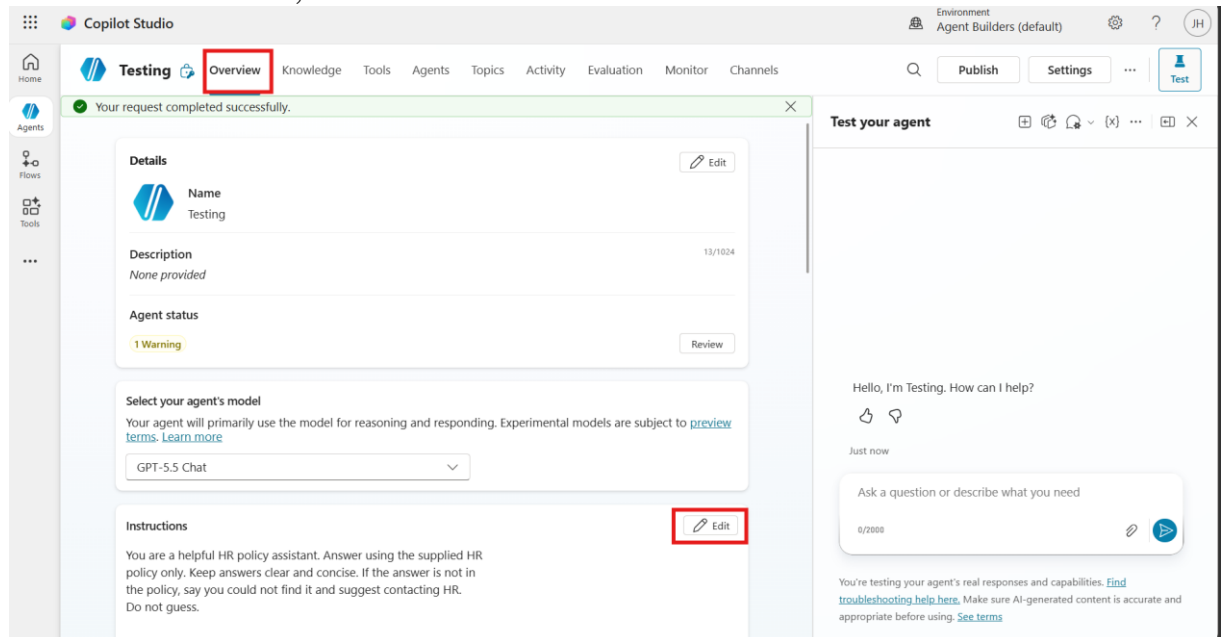
Answers questions from the supplied HR policy.

7. Click **Save**.

✓ **CHECK:** Your agent is created and its name shows at the top of the screen.

Give the agent its instructions

1. On the **Overview** tab, find **Instructions** and click **Edit**.

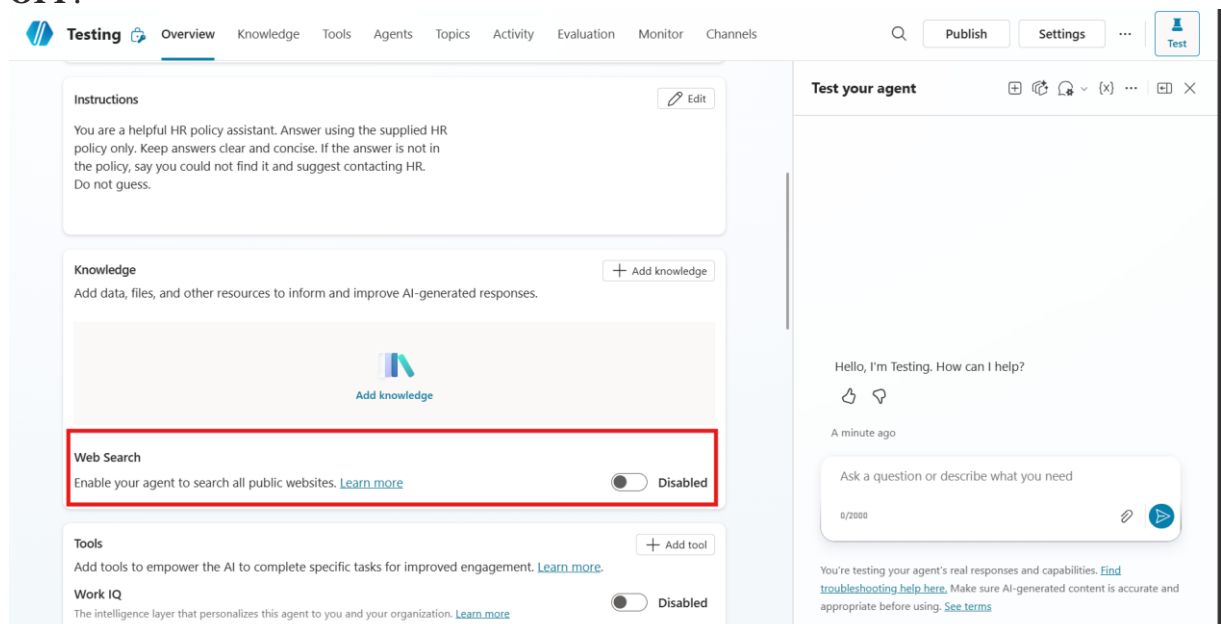


2. Delete anything already in the box, then paste the text below.

Copy this:

You are a helpful HR policy assistant. Answer using the supplied HR policy only. Keep answers clear and concise. If the answer is not in the policy, say you could not find it and suggest contacting HR. Do not guess.

3. Click **Save**.
4. Still on the Overview tab, scroll down to find the **Web search** setting and turn the toggle **OFF**.



⚠ If you leave web search on, the agent will augment its answers from the internet instead of strictly referencing the HR policy.

✓ **CHECK:** The web search toggle is grey and switched off.

Add the HR policy as knowledge

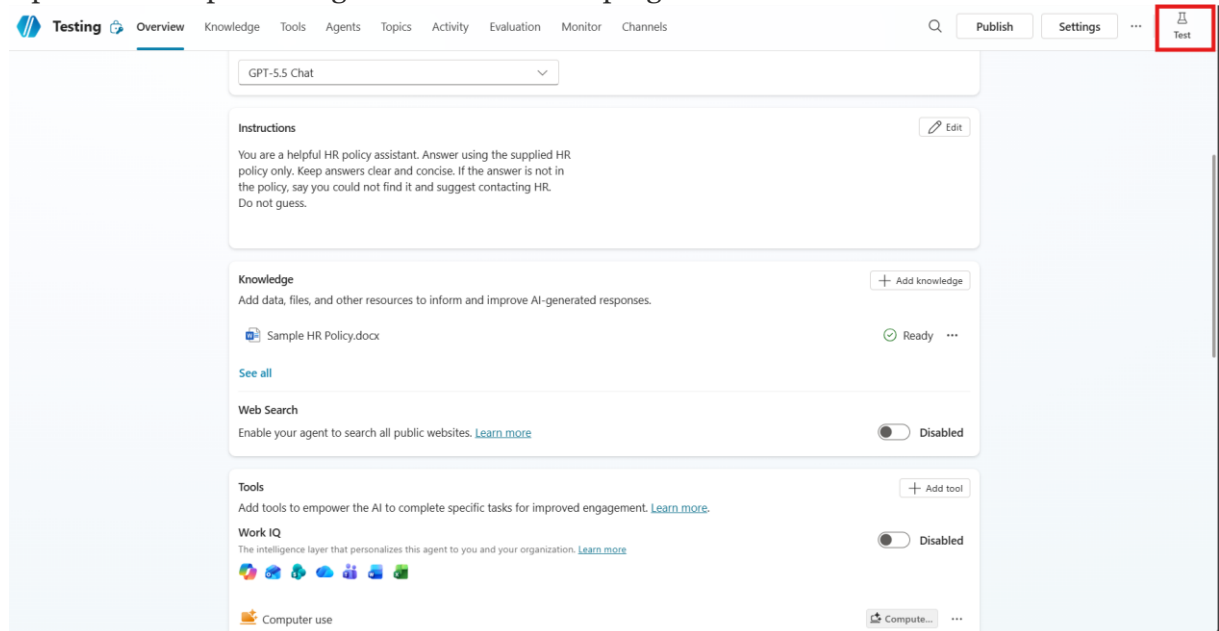
1. Click the **Knowledge** tab along the top of the page.
2. Click **+ Add knowledge**.
3. Click **Browse files from your computer**.
4. Choose the **Sample HR policy** file you downloaded in Setup, then click **Add**.
5. Wait for the file to finish processing. This can take one to two minutes.

The screenshot shows the Agent Builders interface with the **Knowledge** tab selected. The top navigation bar includes **Testing**, **Overview**, **Knowledge**, **Tools**, **Agents**, **Topics**, **Activity**, **Evaluation**, **Monitor**, and **Channels**. The main content area has a dropdown menu set to **GPT-5.5 Chat**. Below this, the **Instructions** section contains a prompt for an HR policy assistant. The **Knowledge** section, highlighted with a red box, shows a file named **Sample HR Policy.docx** with a green checkmark and the word **Ready**. The **Web Search** section has a toggle switch set to **Disabled**. The **Tools** section also has a toggle switch set to **Disabled**. At the bottom, there is a **Computer use** section with a **Compute...** button.

✓ **CHECK:** The knowledge source shows "Ready" with a green tick beside it. Don't carry on until you see the green tick.

Test Agent 1

1. Open the **Test** pane using the button at the top right.



2. Type this question and press Enter:

Copy this:

How do I book annual leave?

✓ **CHECK:** The agent answers using information from the HR policy.

3. Now type a question the policy can't answer, and press Enter:

Copy this:

What is my exact remaining holiday balance?

✓ **CHECK:** The agent says it couldn't find that in the policy and suggests contacting HR. It shouldn't invent a number.

4. If the agent invents an answer, click **Save**, start a new test session, and try again. If it still invents an answer, let us know.

You've now built a simple knowledge-grounded agent, connected it to a document, switched off web search, and tested that it answers from the HR policy rather than guessing. You've also checked the important failure case: when the answer isn't in the document, the agent should say so clearly instead of making something up.

Knowledge grounding is one of the core features of Copilot Studio. In real projects, agents can use a wide range of knowledge sources, including uploaded files, SharePoint sites, Dataverse tables and other enterprise content, so users can ask natural questions and quickly find answers without having to search through documents or systems themselves.

Agent 1 complete.

Agent 2: Build a PDF upload agent

This agent demonstrates how deterministic workflows can be triggered by an agent to perform functions across the Microsoft 365 platform using Tool calling. It takes a PDF from the user, saves it into a SharePoint folder, then posts a notice in a Teams channel. A wide range of third-party connectors are also available to access systems such as HubSpot, Salesforce or DocuSign.

⚠ **This exercise only works with PDF files. Don't try it with a Word document or the agent will fail.**

Create the new agent

1. In the left menu click **Agents**.
2. Click + **New agent**, then **Blank agent**.
3. Wait one to two minutes for it to provision.
4. In **Name**, type:

Copy this:

PDF File Uploader - [your name]

5. In **Description**, type:

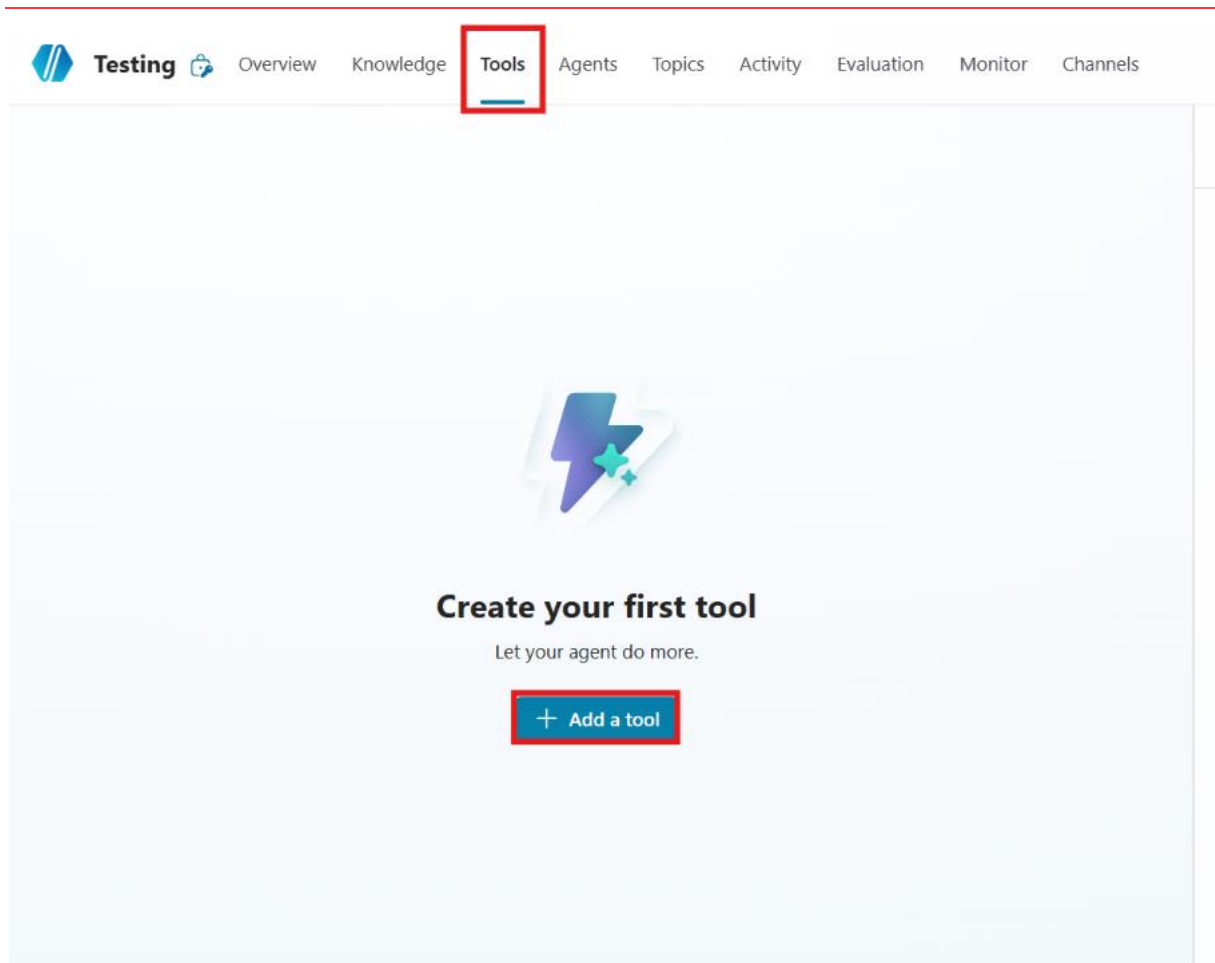
Copy this:

Uploads PDF files to SharePoint and posts a notice in Teams.

6. Click **Save**.

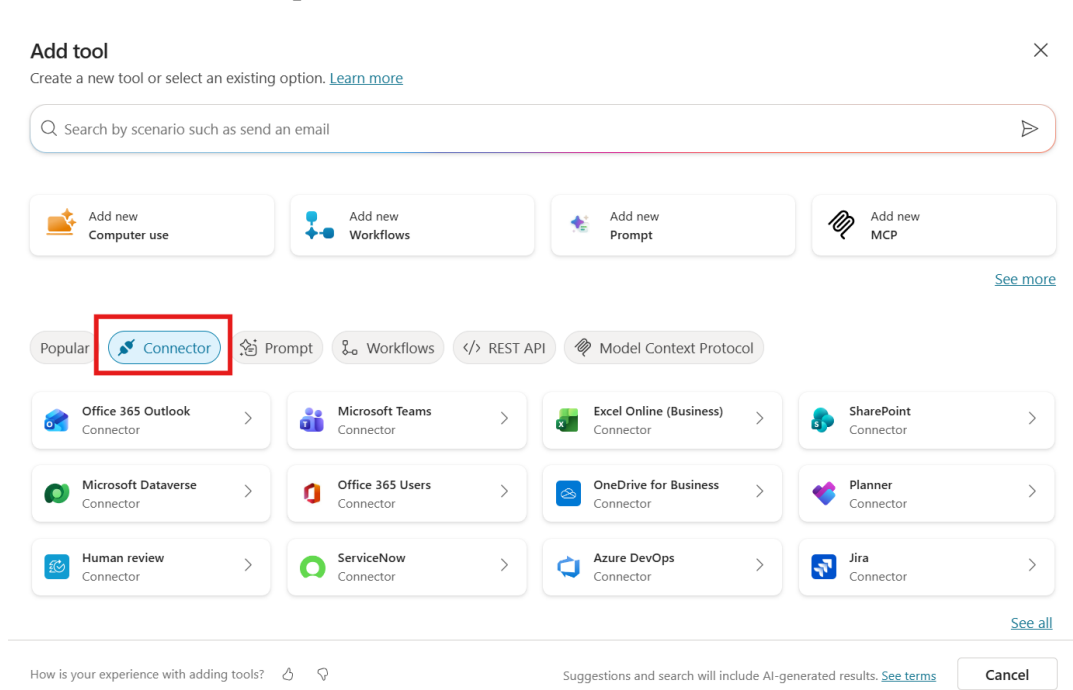
Step A — Add the SharePoint "Create file" tool

1. Click the **Tools** tab along the top of the page.
2. Click + **Add a tool** (shown in the red box below).



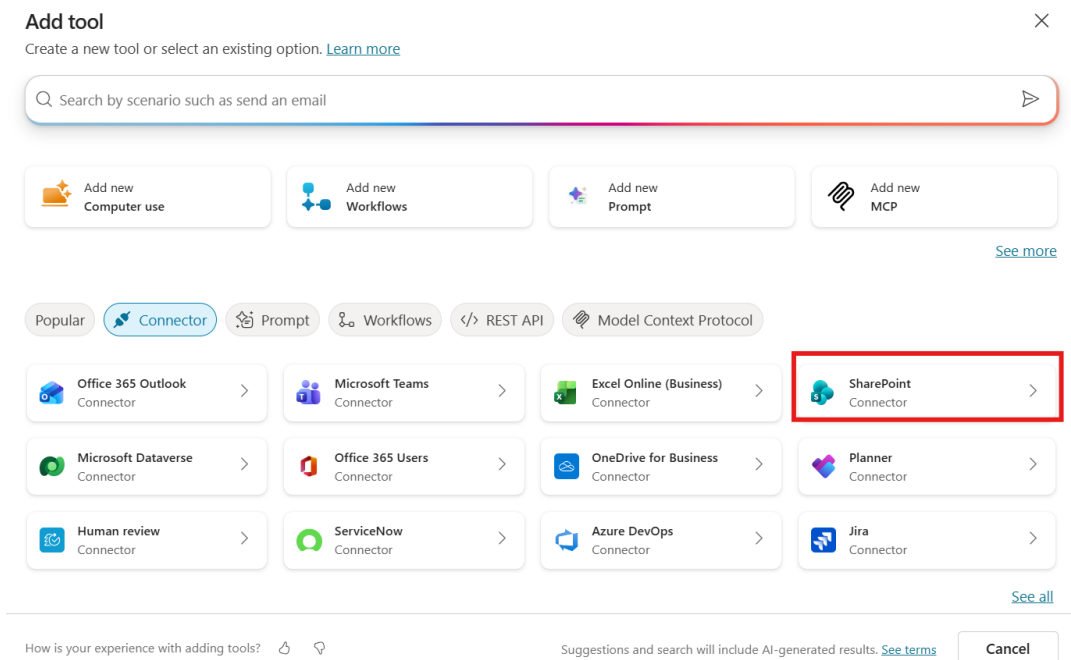
The Tools tab. Click "+ Add a tool".

3. The Add tool window opens. Click the **Connector** tab, shown in the red box below.



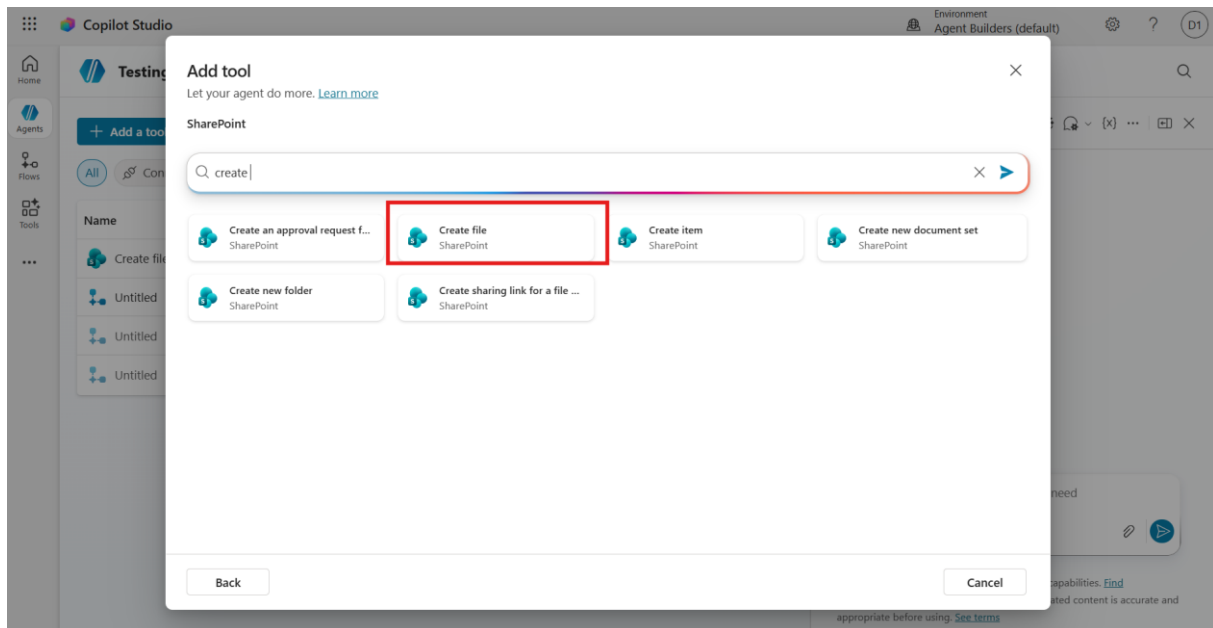
Click the Connector tab.

- Click **SharePoint** in the list, shown in the red box below.



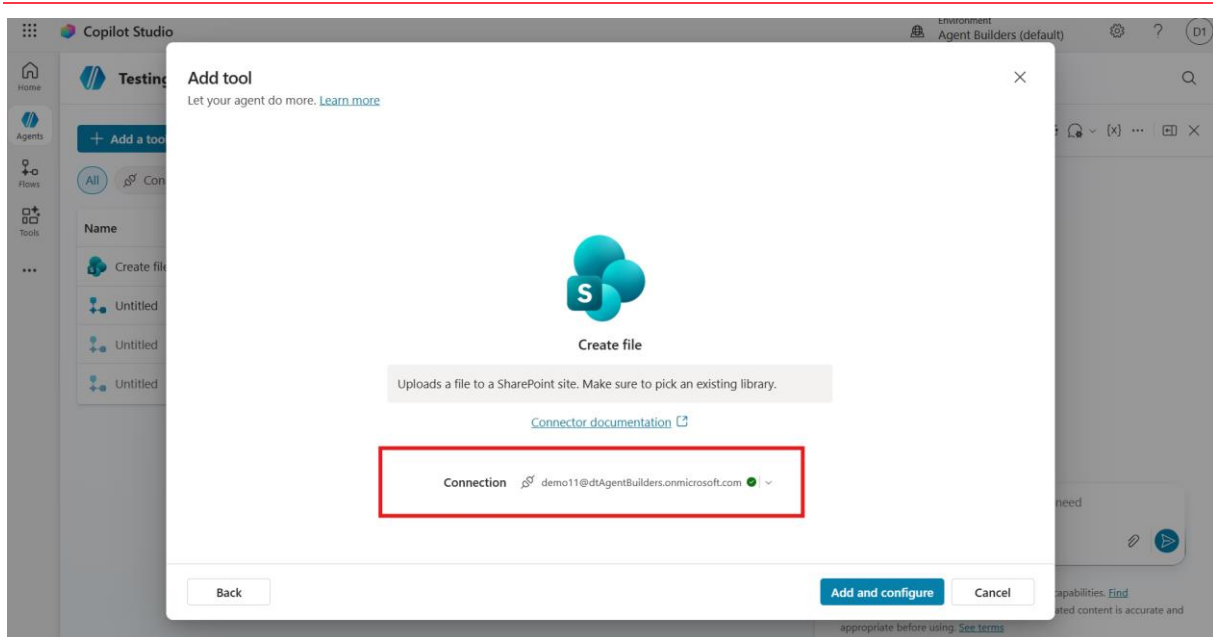
Click the SharePoint connector.

- In the search box type **create**, then click **Create file** in the results, shown in the red box below.



Type "create" in the search box, then choose "Create file".

- Look at the **Connection** row near the bottom of the window, shown in the red box below.
 - If it shows a green tick, go to the next step.
 - If it says "Not connected", click it, choose to create a new SharePoint connection, sign in with your workshop account, and wait for the green tick to appear.



Check the Connection row shows a green tick before continuing.

7. Click **Add and configure**.

Step B — Tell the tool where to save files

This screen's the fiddly one. If you need help just ask.

1. The tool opens. Leave **Name** and **Description** exactly as they are. Don't rename anything here.
2. In the left menu of the tool, click **Inputs**.
3. Find **Site Address**. It might currently say "Dynamically fill with AI". Click it and change it to **Custom value**.
4. Choose **Agent Builders** from the list if it appears. If it doesn't appear, click **Enter custom value** and paste this address:

Copy this:

```
https://dtagentbuilders.sharepoint.com/sites/PivotalEdgeAI
```

⚠ That web address says PivotalEdgeAI, but it is the correct site. The site is displayed as "Agent Builders" and its underlying address is different. This is expected, carry on.

- Find **Folder Path**. Change it to **Custom value**, then click the small **folder icon** at the right of the box.

PDF File Uploader - 16 test Overview Knowledge **Tools** Agents Topics Activity Evaluation Monitor +1

← Create file ▾ Enabled ... **Save**

Details

Inputs ⓘ

Completion

Inputs

What the tool accepts in order to run. Inputs will be filled in the order shown.
[Learn more about inputs for this connector action](#)

Input name	Fill using ⓘ	Value
Site Address * dataset	Custom value	Agent Builders - https://dtanent... Must start with an existing library. Add folders if needed.
Folder Path * folderPath	Custom value	SharePoint _catalogs > _cts > _private > _vti_pvt > - FormServerTemplates > - images > - Lists > - Shared Documents > - SiteAssets > - SitePages > - Style Library >
File Name * name	Dynamic	
File Content * file	Dynamic	

Completion

Specify what your agent does when it finishes

After running:

Don't respond (default)

> **Advanced**

6. Choose the document library **Shared Documents**, then choose the folder **Testing**.

PDF File Uploader - 16 test Overview Knowledge Tools Agents Topics Activity Evaluation Monitor +1

← Create file ▾ Enabled ☒ ... Save

Details

Inputs ⓘ

Completion

Inputs

What the tool accepts in order to run. Inputs will be filled in the order shown.
[Learn more about inputs for this connector action](#)

Input name	Fill using ⓘ	Value
Site Address * dataset	Custom value	Agent Builders - https://dtagent- Must start with an existing library. Add folders if needed.
Folder Path * folderPath	Custom value	SharePoint > Shared Documents Forms Sample Docs Testing
File Name * name	Dynamically fill with AI	
File Content * file	Dynamically fill with AI ▾	Customize

Completion

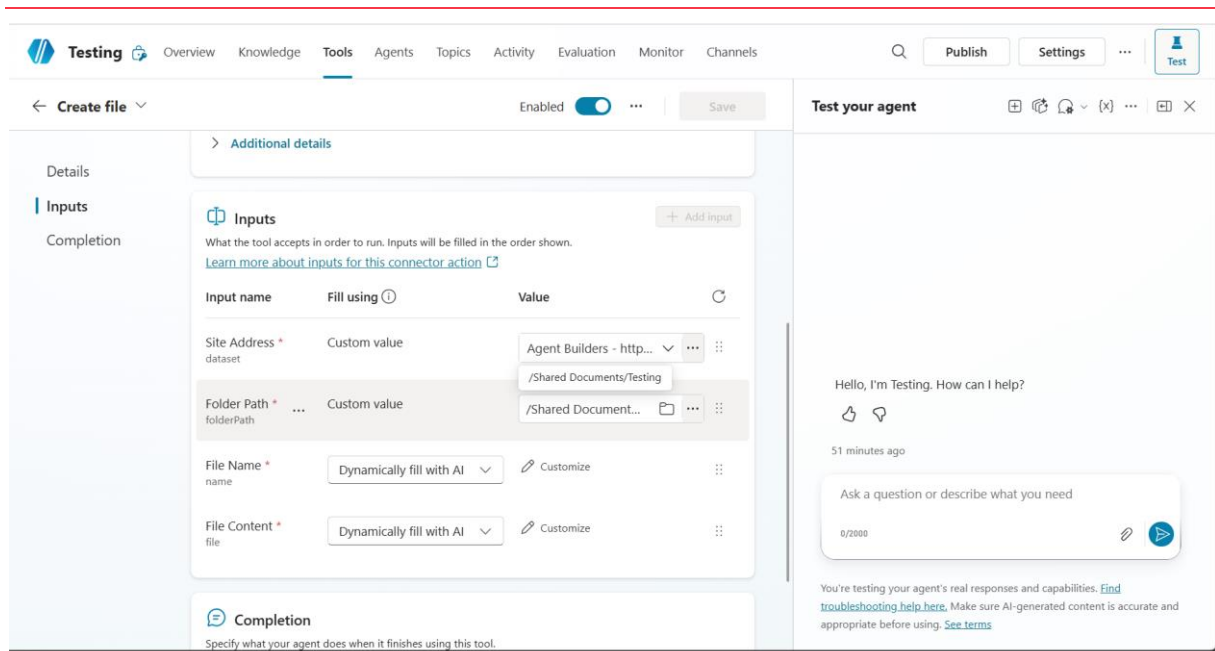
Specify what your agent does when it finishes using this tool.

After running:

Don't respond (default) ▾

> Advanced

7. If you can't click a folder icon, then click custom value and paste this.
/Shared Documents/Testing
8. Leave **File Name** and **File Content** set to **Dynamically fill with AI**. Don't change these two.
9. Click **Save**.



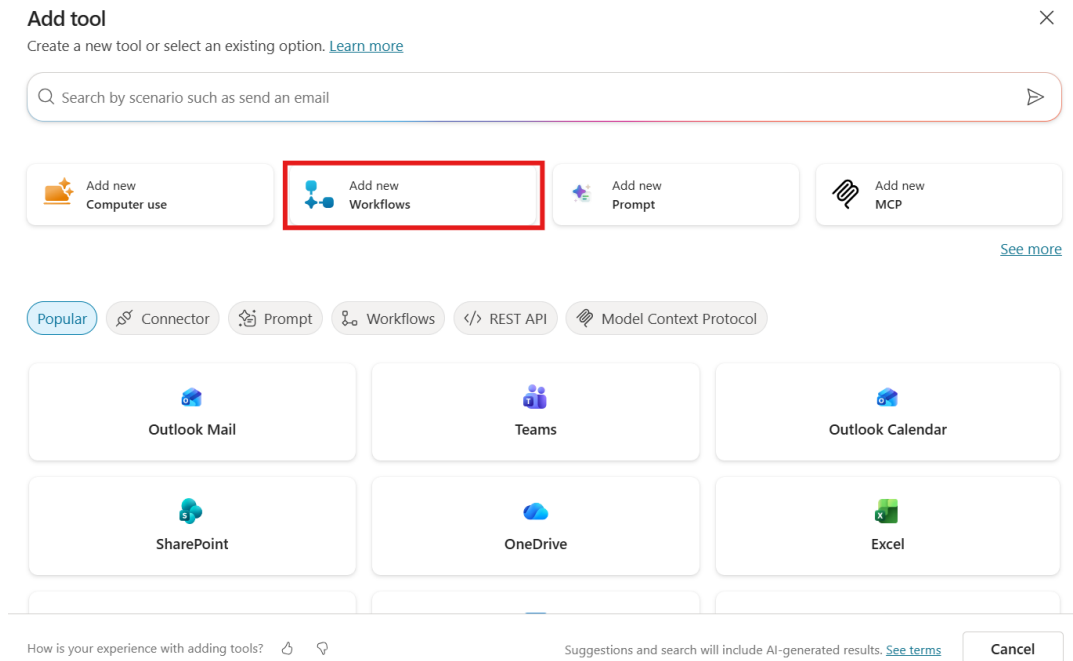
The Inputs screen. Site Address and Folder Path are set to Custom value; File Name and File Content stay on Dynamically fill with AI.

✓ **CHECK:** Site Address shows the Agent Builders site, and Folder Path ends in /Shared Documents/Testing.

Step C — Build the Teams notification flow

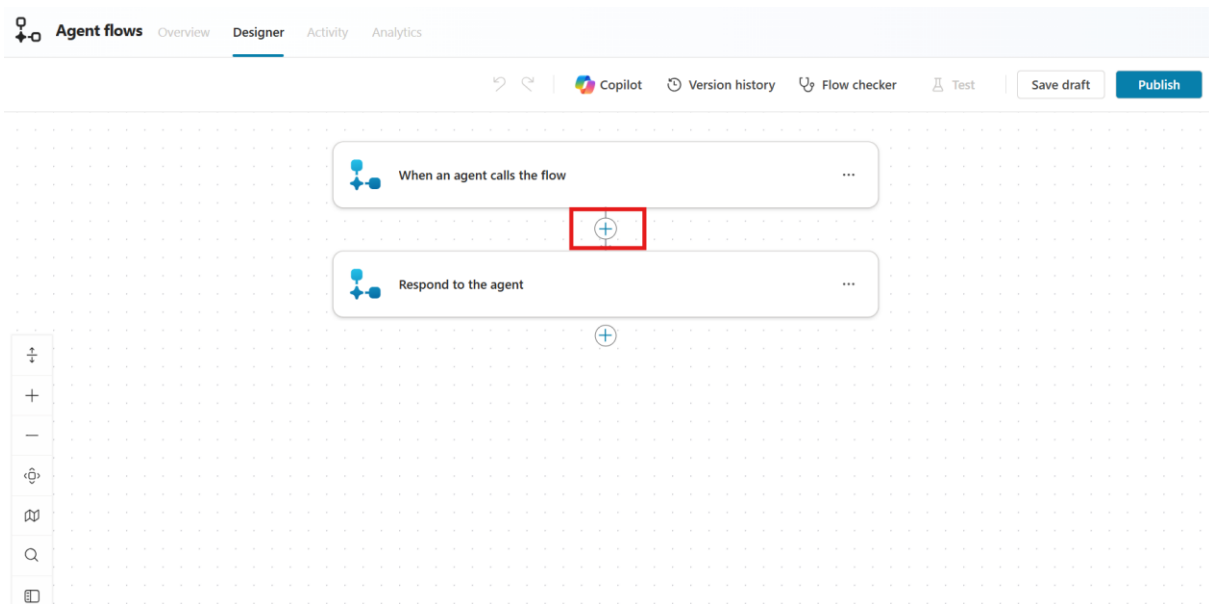
1. Click the **Tools** tab again.
2. Click + **Add a tool**.
3. At the top of the window, click the **Add new Workflows** tile, shown in the red box below.

⚠ **This tile is at the top of the Add tool window, above the connector list. Microsoft renames it from time to time. If it doesn't say "Workflows", look for the tile with the same branching-arrow icon.**



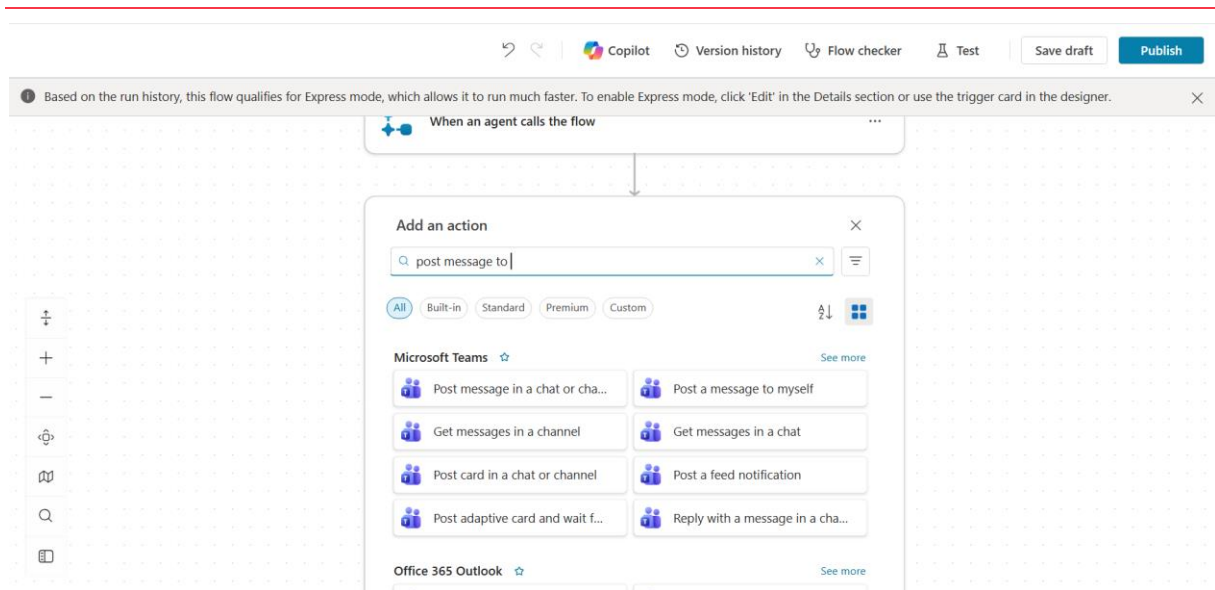
Click the "Add new Workflows" tile at the top.

4. Wait for the flow designer to open. You'll see two boxes: **When an agent calls the flow** and **Respond to the agent**.
5. Click the small + circle in between those two boxes, shown in the red box below.



Click the + between the two existing boxes, not the one underneath.

6. In the search box type **post message**.
7. Under **Microsoft Teams**, click **Post message in a chat or channel**.



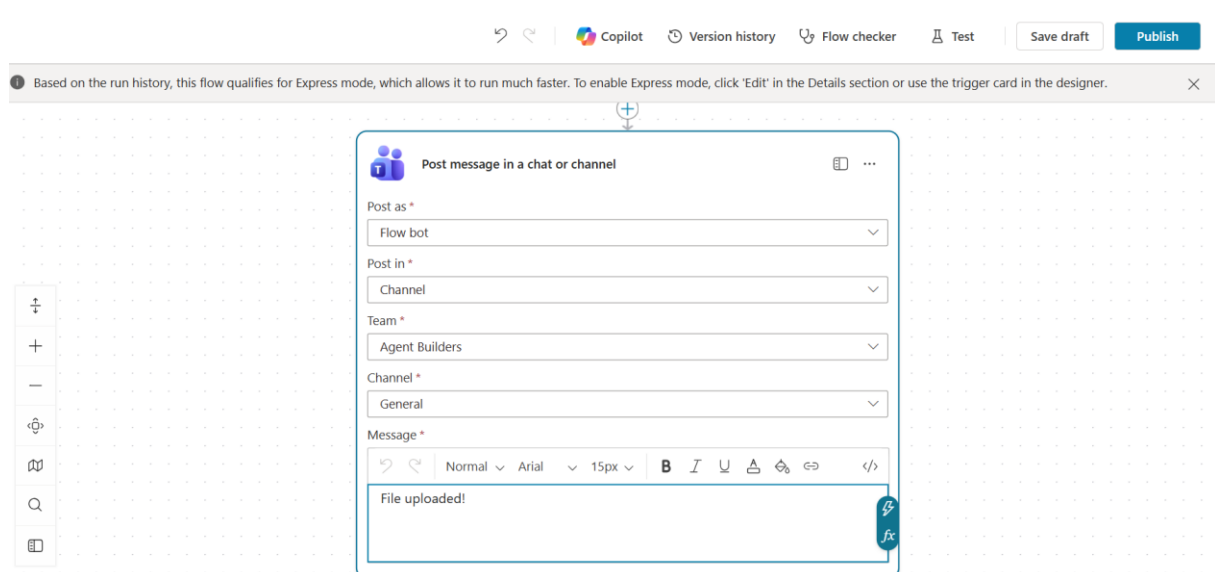
Choose "Post message in a chat or channel" under Microsoft Teams.

8. If it asks you to sign in, click **Sign in** and choose your workshop account in the pop-up window.
9. Now fill in the five boxes exactly as follows:
 - Post as → **Flow bot**
 - Post in → **Channel**
 - Team → **Agent Builders**
 - Channel → **General**
 - Message → type the text in the box below

Copy this:

File uploaded! - [Your name]

⚠ Type the message exactly as " File uploaded! - [Your name] ". Later steps look for this exact text, and your final check will fail if it doesn't match.



All five boxes filled in, with the message reading "File uploaded!".

10. Click **Save draft** at the top right.
11. Click **Overview** at the top left.
12. In the **Details** panel click **Edit**, and change the flow name to:

Copy this:

File uploaded - [your name]

13. Click **Save**.
14. Click **Designer** to go back, then click **Publish** at the top right.
15. Go to tools and make sure it looks like this. You should see the flow and the SharePoint tool.

PDF File Uploader - 16 test Overview Knowledge Tools Agents Topics Activity Evaluation Monitor +1

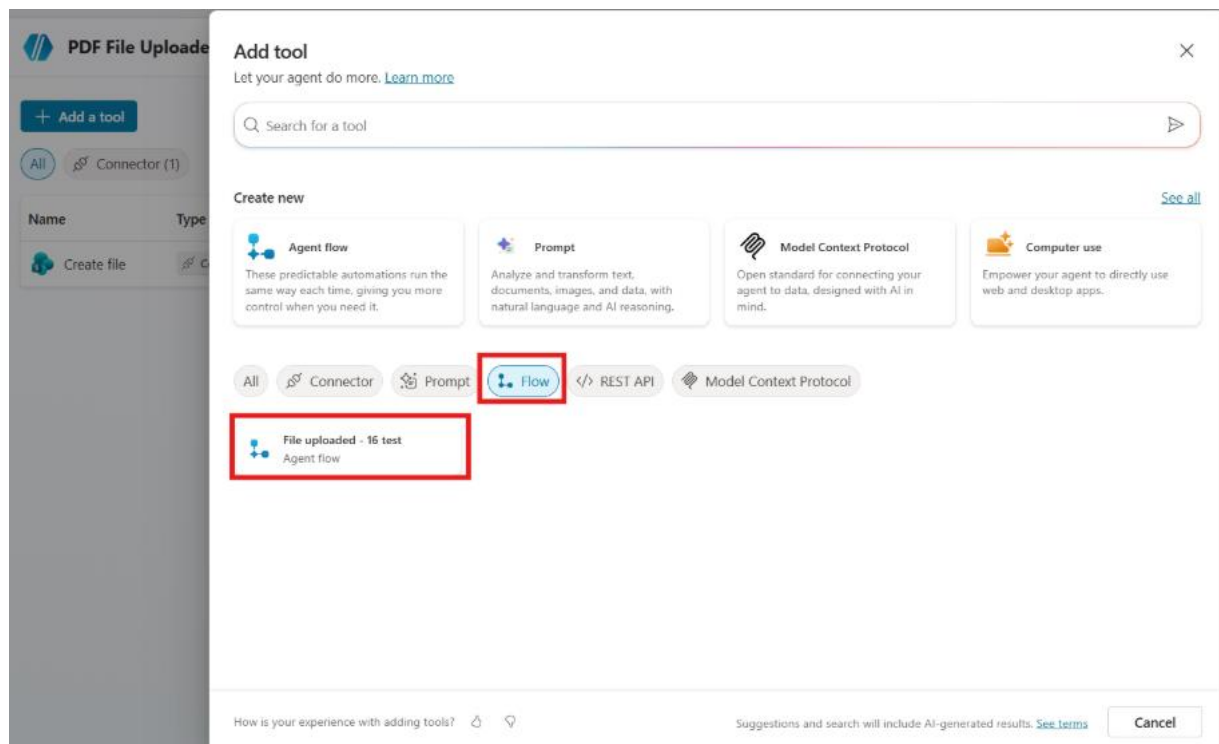
+ Add a tool Search tools

All Connector (1) Flow (1) Last refreshed now

Name	Type	Available to	Trigger	Last modified	Errors	Blocked	Enabled
File upload...	Flow	PDF File Up...	By agent	Workshop Demo 16 1 ...			On
Create file	Connec...	PDF File Up...	By agent	Workshop Demo 16 11 ...			On

dio.microsoft.com/environments/.../details

16. If not then add the flow under tools -> **add a tool** – make sure it is published otherwise it will not show up here.



✓ **CHECK:** The flow says it has been published. An unpublished flow won't run.

Step D — Tell the agent to use the tool and the flow

1. In the left menu click **Agents**.
2. Find and click your **PDF File Uploader - [your name]** agent.
3. On the **Overview** tab find **Instructions** and click **Edit**.
4. Delete anything in the box and paste the text below.

Copy this:

When a user attaches a PDF, first use the Create file tool to upload it to SharePoint. After the upload succeeds, call your File uploaded flow to post a message in the Agent Builders General channel in Teams. Only confirm success to the user after both actions have succeeded. If the file is not a PDF, explain that this agent accepts PDFs only and do not call either tool.

5. Click **Save**.

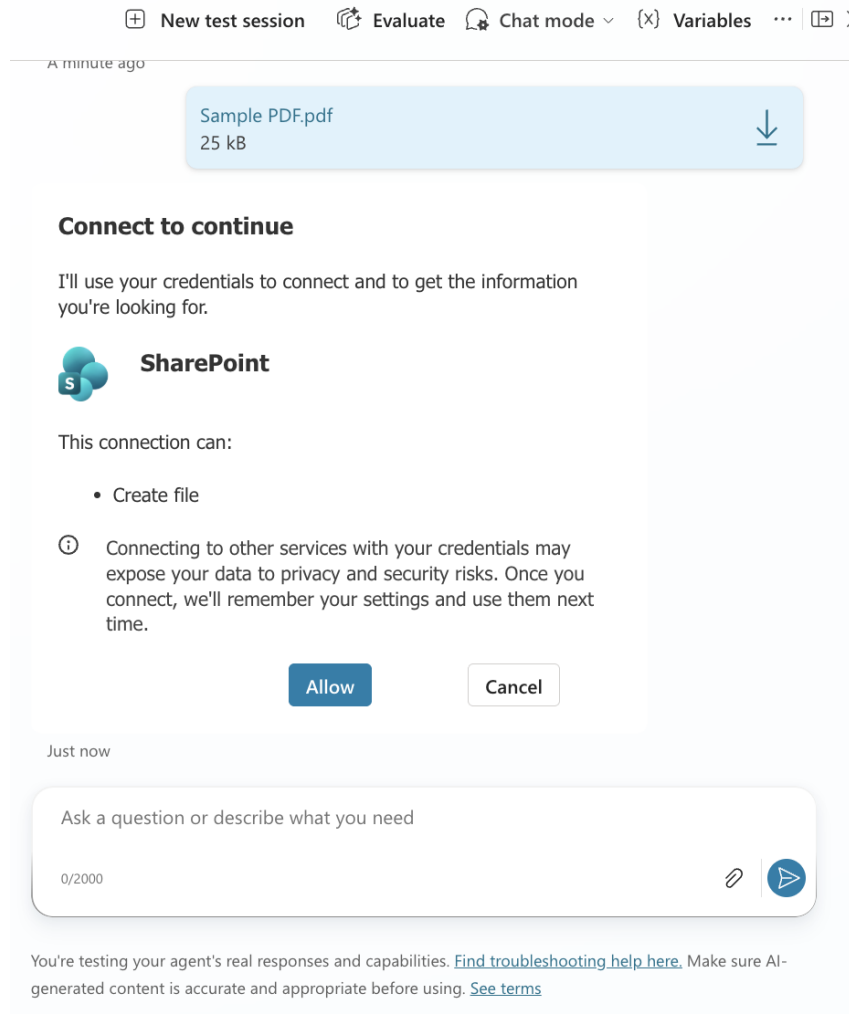
Step E — Test Agent 2

1. Go back to the **My Apps** browser tab you left open in Setup.
2. Find **Teams** in the app list and click it. Sign in if you are asked. Leave this tab open, you'll check it in a minute.
3. Go back to the **Copilot Studio** tab and open the **Test** pane.
4. Click the **paperclip** icon at the bottom of the test pane and attach **Sample PDF.pdf** from your Downloads folder.
5. Type this message and press Enter:

Copy this:

Please upload this PDF

6. A **Connect to continue** message will appear for SharePoint. Click **Allow**.



When the "Connect to continue" message appears, click Allow.

7. A second **Connect to continue** message may then appear for Teams. Click **Allow** again.
8. Wait for the agent to reply. It should confirm that it has uploaded the file and posted the message.

- ✓ **CHECK:** The PDF now appears in SharePoint, in Shared Documents > Testing.
- ✓ **CHECK:** The message "File uploaded!" appears in Teams, in the Agent Builders team, General channel.

You've now built an agent that can trigger a deterministic workflow across Microsoft 365. It accepts a PDF from the user, uses a SharePoint connector to save the file in the right folder, then calls a Teams workflow to post a notification in the correct channel. You've also seen how connector permissions, tool configuration and agent instructions work together so the agent can carry out a reliable, repeatable process rather than just answer questions.

A wide range of connectors is available for both the Microsoft platform and 3rd party systems.

Agent 2 complete.

Agent 3: Build a computer-use agent

This agent demonstrates how Computer Use can help with accessing information or carrying out actions on applications and website. This can be useful where an API isn't available, particularly with legacy apps or websites. The Computer Use tool opens a website, looks at the screen, clicks on pages and links and retrieves information, the same way a user would.

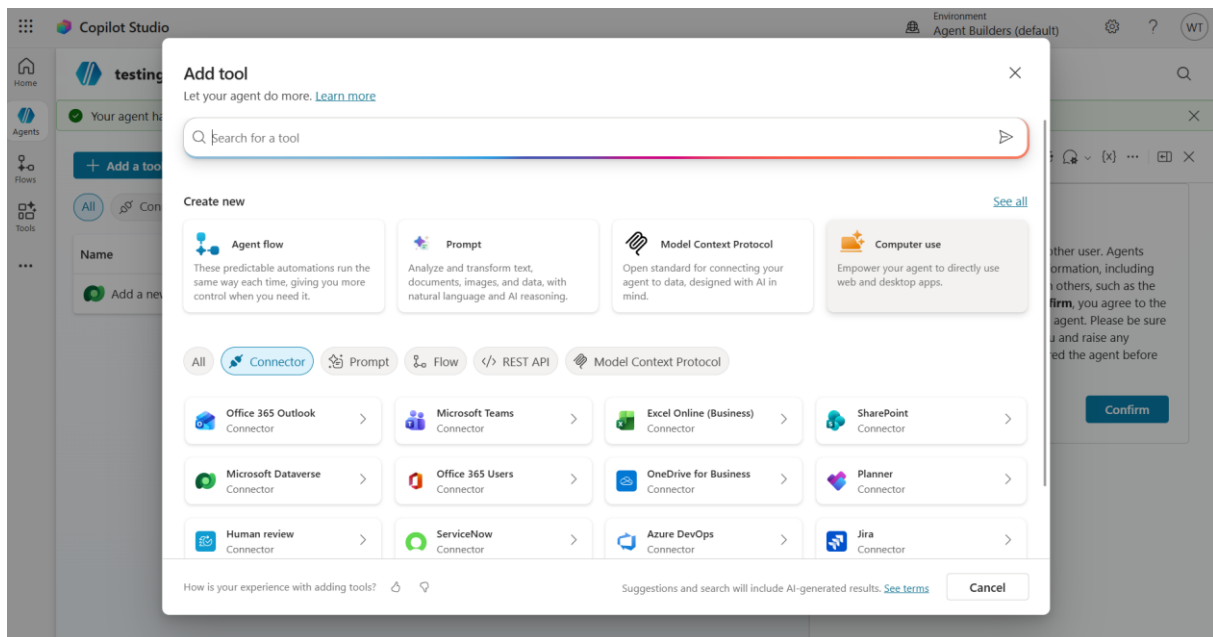
Create the agent

1. In the left menu click **Agents**, then **+ New agent**, then **Blank agent**.
2. Wait one to two minutes for it to provision.
3. In **Name**, type:

Copy this:

Computer Use - [your name]

4. Click **Save**.
5. Click the **Tools** tab, then **+ Add a tool**.
6. At the top of the window, click the **Add new Computer use** tile.



The "Computer use" tile sits along the top row of the Add tool window.

7. Wait for the tool to open.

Set the tool up

1. Paste instructions into tool

Copy this:

Open <https://agentbuilders.pro>
If a cookie message appears, click Accept.
Click the link that says "Join our next event".
Read the page and tell me what events are coming up and what the group is about.
Do not sign in, do not download anything, do not fill in any forms, and do not leave agentbuilders.pro.

2. Click **Add and configure**.

**New computer use**×

Empower your agent to directly use web and desktop apps. [Learn more](#)

Instructions

Describe what you want the computer use to do

For example:

1. Open <https://example.com>
2. Go to Add Invoice
3. Fill in the provided invoice details
4. Select Submit Invoice

0/8000

Or start with prompt template

Invoice processing

Extracts invoice data from a PDF and enters it into a web form.

Data entry

Submits multiple products into an inventory form.

Data extraction

Looks up specific information on a website.

Back

Add and configure

Cancel

3. In the left menu of the tool, click **Details**.
4. Find the **Description** box and type this in. The agent reads this description to work out when to use the tool, so it matters.

Copy this:

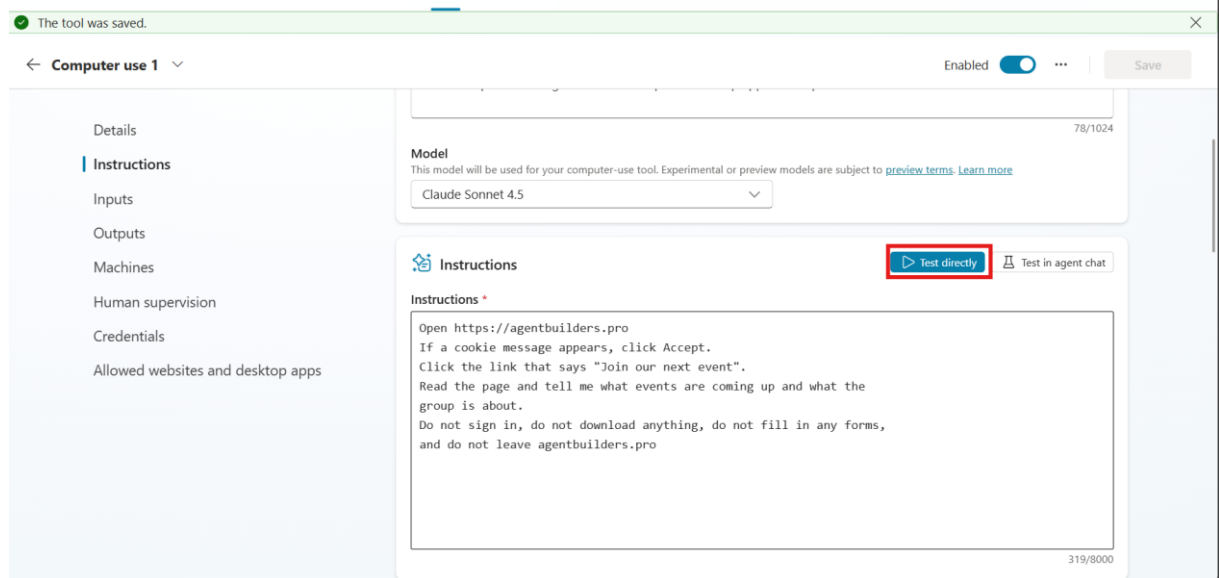
Use this tool to look up the latest Agent Builders events from agentbuilders.pro. Call it whenever someone asks about the next event, upcoming meet-ups, or what the group has coming up.

5. Click **Save**.

Test the tool on its own

First check the tool works by itself, before you let the agent loose on it.

1. Click **Test directly** at the top of the tool.



2. A preview window opens showing a live browser. It takes 30 to 60 seconds to start.
3. Watch what it does. Don't click inside the preview window while it is working.
4. Wait until it comes back with info about the upcoming Agent Builders events

✓ **CHECK:** The agent opens agentbuilders.pro, accepts the cookie message, clicks "Join our next event", and then tells you what events are coming up.

Now test it through the agent

The tool works. Now check the agent knows when to reach for it on its own.

1. Open the **Test** pane at the top right.

2. Type this and press Enter:

The screenshot shows the Agent Builders interface. The top navigation bar includes 'Overview', 'Knowledge', 'Tools', 'Agents', 'Topics', 'Activity', 'Evaluation', and '+2'. The 'Tools' section is active, showing a tool named 'Computer use 1'. The 'Details' section includes a 'Name' field with 'Computer use 1', a 'Description' field with 'Use this tool to look up the latest Agent Builders events from agentbuilders.pro. Call it whenever someone asks about the', and a 'Model' dropdown set to 'Claude Sonnet 4.5'. The 'Instructions' section includes a 'Test directly' button and a 'Test in agent chat' button. The 'Test your agent' pane on the right shows a chat conversation where the user asks 'When is the next Agent Builders event?' and the agent responds with 'Hello, I'm Computer Use - Geoff. How can I help?'.

Copy this:

When is the next Agent Builders event?

3. Notice you didn't tell it to use the tool. The agent orchestration in Copilot takes your prompt and works out which tool is the right one, and runs it.
4. Watch it work in the test pane. Note, the screenshots you see are just for the test pane, end users of a deployed agent won't see these.

✓ **CHECK:** The agent calls the computer use tool on its own, and comes back with the event details.

Agent 3 complete.

Where you'd actually use this

Computer use works well when a system has no API and no integration and you need to do simple data retrieval or perform basic steps. If there's a documented API, use that instead. It'll be faster and more reliable.

The classic use case is the horrible old internal system you have to log into every morning. It was built thirty years ago, the interface is dreadful, there's no API, and nobody is ever going to rewrite it. You point a computer-use agent at it with stored credentials, let it log in, find the report, download it, and drop the file somewhere useful like SharePoint. Put that on a daily schedule and it's already done by the time you get in.

Other things people use it for:

- Supplier and partner portals that were never built to integrate with anything.
- Pulling a report out of a system that only exports through the screen.

-
- Filling in the same web form over and over.
 - Watching a page and flagging it when something changes.

⚠ Two things worth remembering. Credentials belong in a proper stored connection, never typed into the instructions – Copilot has features to securely manage these credentials. And screen automation is slower and more brittle than an API, so use it when there isn't a better option, not first.

Creative challenge

Now make it your own. Change the description and the instructions, point it at something else, and see what happens. The most interesting one gets shown to the group at the end.

Some ideas:

- Pick a public site you actually use and have it pull one thing out. Train times, the weather, a league table, the top headline.
- Have it fill in a form and report back what happened.
- Point it at a simple browser game and see how far it gets.
- Give it something that takes several clicks, then watch where it goes wrong. That's usually more interesting than one that works first time.
- Write a deliberately vague instruction and see what it makes of it.

Worth knowing while you experiment:

- The description is what makes the agent decide to call the tool. Change the description and you change when it fires.
- Small changes in wording make a big difference. If it goes the wrong way, tighten the instruction rather than piling on more of them.
- Rename the agent so it still matches what it does.

If you get stuck

Problem	What to do
My screens don't match this guide	Check the top right says Agent Builders (default). If a banner offered the "New Copilot Studio experience" and you clicked Try now, switch back to the old experience.
My agent won't finish provisioning	Wait two full minutes without refreshing. If it is still spinning, refresh once. If it still fails, let us know.
The knowledge source won't turn green	Wait. Large files take a minute or two. Then click Save and start a new test session. Check you uploaded the HR policy and not the PDF.
My agent invents answers	Check web search is turned off, and that the instructions text was pasted in full including "Do not guess."
The file did not arrive in SharePoint	Check the Create file tool: Site Address and Folder Path must both be set to Custom value, pointing at Shared Documents > Testing.
No message appeared in Teams	Check the flow was Published, not just saved as a draft. Check the message text is exactly "File uploaded!". Check the connection has a green tick.
The computer-use agent won't open the site	Check the address starts with https://. The tool will only open secure sites. Also check the description and instructions saved.
Something asks me to change a password or permission	Stop and let us know. Don't change it.

⚠ Still stuck? Let us know. Don't change any password, permission, connection or setting to try to fix it yourself.