

Uplivra is in beta and online purchasing is paused. **Apply to test it:** testers get a license on request and 10% off when purchasing opens.

INSTALL GUIDE · ADMINISTRATORS

# Connect an AI

Optional and free. Three ways to use an AI with Uplivra - let an AI app your company already uses read Uplivra, point Uplivra at a model server on your own network, or use a cloud AI service with your own API key - and what each one sends.

Uplivra Technologies LLC · Guide for Uplivra 26.10 · Updated 4 October 2026 · Latest version:  
<https://uplivra.com/guides/connect-ai.html>

## Before you start

Uplivra doesn't need an AI. **Insights** (questions in plain words, likely causes, notices, Running out forecasts) is Uplivra's own analysis and works without one. Uplivra ships no AI model and there is nothing extra to download.

Connecting an AI is optional and free from Uplivra. Everything below is under **Settings > Connect an AI**, and only an administrator can change it. There are three ways:

| Way                      | What it does                                                                                       | Does data leave your network?         |
|--------------------------|----------------------------------------------------------------------------------------------------|---------------------------------------|
| 1. Connect your own AI   | An AI app your company already uses (Claude, Microsoft Copilot or another MCP app) reads Uplivra   | Only if that AI app runs in the cloud |
| 2. Your own model server | A model on your network words the answers on <b>Insights &gt; Ask</b>                              | No                                    |
| 3. A cloud AI service    | OpenAI, Anthropic (Claude), Azure OpenAI or Google Gemini words the answers, with your own API key | Yes: see <a href="#">What is sent</a> |

You can use way 1 together with way 2 or way 3.

## Way 1: connect your own AI

Your AI app connects to Uplivra over **MCP** (Model Context Protocol, a standard way for AI apps to read other software). The connection is **read-only**: Uplivra gives the AI no way to change devices, settings or alerts. It is **off** until an administrator turns it on.

### Turn it on

- 1. Open **Settings > Connect an AI**.
- 2. Tick **Allow other AIs to read Uplivra** and click **Save**.

### Make a connection (one per AI app, or one per person)

- 1. Under **Add an AI**, pick **Which AI**: Claude Code or Claude Desktop, Claude (claude.ai Team or Enterprise), Microsoft Copilot Studio, Google Gemini (Antigravity), Open WebUI, or **Other MCP app**.
- 2. Tick **What it may read**: sites and collectors, devices and their checks, alerts, changes, log messages (syslog), SNMP traps, and questions to Insights. Sites, devices, alerts and changes are ticked to start with. Log messages can contain user names and addresses, so tick them only if you need them.
- 3. Give it a **Name** (for example *Copilot for the help desk*) and choose how long it **works for**: 7 days, 30 days, 90 days or 1 year.
- 4. Click **Create the connection**.
- 5. **Copy the token now**. Uplivra keeps only a fingerprint of it, so it can't show it again. Keep it like a password.
- 6. Follow the steps Uplivra shows for the AI you picked. Most apps need these three things:

| Setting   | Value                                        |
|-----------|----------------------------------------------|
| Address   | https://<your Uplivra server>/mcp            |
| Transport | Streamable HTTP                              |
| Sign-in   | Request header Authorization: Bearer <token> |

For Claude Code, Uplivra shows a ready command to paste:

```
claude mcp add --transport http uplivra https://<your Uplivra server>/mcp --header
"Authorization: Bearer <token>"
```

7. Ask your AI something like *"What's down in Uplivra right now?"*

### Good to know

- **Each token belongs to the person who made it.** The AI never sees more than that person can see in Uplivra, and only what was ticked.
- **AI apps that run in the cloud** (claude.ai, Copilot Studio) must reach your Uplivra server from the internet over HTTPS, with a trusted certificate. Apps on your own computers (Claude Code, Claude Desktop, Open WebUI) only need your network.
- **ChatGPT** is in the list but can't connect yet: it signs in to MCP servers in a different way (OAuth) that Uplivra doesn't offer yet.
- Each connection can ask up to 600 questions an hour.
- **Every call is in the audit log** (Settings > Audit log, action `mcp.call`).
- **To stop one AI**, click **Disconnect** next to it under **Connected AIs**. It stops right away.
- Customers' own administrators make connections; an MSP can't make them for a customer.

## Way 2: your own model server

A model server is a program on one of your computers that runs an AI model. Uplivra uses it to **word the answers** on **Insights > Ask** in full sentences, and to **read unusually worded questions**. Uplivra still finds the facts itself; the model only works with what Uplivra found. **Nothing leaves your network.**

Any server with an OpenAI-style `/v1/chat/completions` address works, for example **Ollama**, **LM Studio**, **llama.cpp's server**, **vLLM** or **LocalAI**. These steps use Ollama, the simplest.

### On the computer that will run the model

1. Pick the computer: this Uplivra server or another computer on your network. 16 GB of memory runs an 8-billion-parameter model; a graphics card (GPU) makes answers much faster.
2. Install Ollama from [ollama.com/download](https://ollama.com/download).
3. Download a model. Open a terminal (Command Prompt or PowerShell on Windows) and run `ollama pull llama3.1:8b`. When it finishes, `ollama list` shows the model's exact name.

- Only if Ollama is on **another** computer than Uplivra: set the setting `OLLAMA_HOST` to `0.0.0.0` (on Windows: *Edit the system environment variables*, then restart Ollama), and allow TCP port **11434** in that computer's firewall. Give that computer a fixed IP address so the address below doesn't change.

## In Uplivra

- Open **Settings > Connect an AI**. Under **Where the model is**, choose **Your own model server**. The page shows these same steps.
- Address:** `http://<that computer's IP>:11434`, for example `http://192.168.1.20:11434`. If Ollama runs on the Uplivra server itself: `http://127.0.0.1:11434`. It must be a private address; Uplivra refuses addresses on the internet here.
- Model:** exactly as `ollama list` shows it, for example `llama3.1:8b`.
- API key:** leave it empty (Ollama doesn't use one).
- Choose what the model may do: help read the question, write the answer, and whether it may see log message text (passwords and keys are masked either way).
- Tick **Use an AI model** at the top and click **Save and test the model**. Uplivra sends a one-word test and shows how long the answer took.

**LM Studio instead of Ollama:** in LM Studio's *Developer* tab, load a model, turn on *Serve on local network* and click *Start server*. The address is `http://<IP>:1234`; the model name is the one LM Studio shows.

**Tip:** A model on a computer without a graphics card works, but can take 30 seconds or more per answer. Raise **Wait for the model up to** if the test times out.

## Way 3: a cloud AI service

Uplivra can send questions to **OpenAI**, **Anthropic (Claude)**, **Azure OpenAI** or **Google Gemini**, using an API key from your own account with that company. You pay the provider directly; Uplivra charges nothing for it. It is **off** by default.

Before you start: the Uplivra server must reach the provider over the internet (HTTPS, port 443). If it goes out through a web proxy, set `HTTPS_PROXY` for the Uplivra service.

## Get a key from your provider

### OpenAI

1. Sign in at [platform.openai.com](https://platform.openai.com) with your company's account. Under **Settings > Billing**, add a payment method or credit.
2. Open **API keys**, click **Create new secret key**, name it *Uplivra* and copy the key (it starts with `sk-`). It's shown only once.
3. Model to use: `gpt-4.1-mini` (low cost, good enough for this). Address: leave empty.

### Anthropic (Claude)

1. Sign in at [console.anthropic.com](https://console.anthropic.com) with your company's account. Under **Settings > Billing**, buy credits.
2. Open **API keys**, click **Create key**, name it *Uplivra* and copy the key (it starts with `sk-ant-`). It's shown only once.
3. Model to use: `claude-haiku-4-5` (fast, low cost). Address: leave empty.

### Azure OpenAI

1. In the Azure portal, create an **Azure OpenAI** resource (or use one you have), in a region your company allows.
2. Open it in Azure AI Foundry and **deploy a model**, for example `gpt-4.1-mini`. Write down the **deployment name** you give it.
3. Back on the resource, open **Keys and Endpoint**. Copy **KEY 1** and the **Endpoint** (like `https://acme.openai.azure.com`).
4. In Uplivra, the **Address** is the endpoint and the **Model** is the deployment name (not the model's name).

### Google Gemini

1. Sign in at [aistudio.google.com](https://aistudio.google.com) with your company's Google account. Click **Get API key > Create API key**, pick or create a Google Cloud project, and copy the key.
2. Turn on billing for that project. On Google's free tier, Google may use what you send to improve its products, so don't use it with company data.
3. Model to use: `gemini-2.5-flash`. Address: leave empty.

### Enter it in Uplivra

1. Open **Settings > Connect an AI**. Under **Where the model is**, choose your provider. The page shows that provider's steps.
2. **Address**: empty, except for Azure OpenAI (your endpoint).

3. **Model:** the model name above (for Azure OpenAI, the deployment name).
4. **API key:** paste it. It is stored encrypted on your Uplivra server and never shown again. Switching to another provider removes the saved key.
5. Read the warning and tick **I understand what is sent and want to use this cloud service**.
6. Leave **Send device names, site names and addresses as they are** unticked (recommended): Uplivra then replaces them with stand-ins such as `host-3` before sending, and puts the real names back in the answer you see.
7. Tick **Use an AI model** at the top and click **Save and test the model**.

## If the test fails

| Uplivra says                                         | What to do                                                                                                                                                                        |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The AI service didn't accept the key                 | Copy the key again, with no spaces. For Azure OpenAI, use KEY 1 from the same resource as the endpoint.                                                                           |
| The AI service doesn't know that model or deployment | Check the model name's spelling. For Azure OpenAI, use the deployment name. For Ollama, use the name <code>ollama list</code> shows.                                              |
| Too many requests or the account is out of credit    | Add credit or billing with the provider, or wait a minute and test again.                                                                                                         |
| Nothing answered at that address                     | Your own model server isn't running, or isn't listening on the network (Ollama: <code>OLLAMA_HOST=0.0.0.0</code> ), or its firewall blocks the port.                              |
| The AI model took too long to answer                 | Use a smaller model or a computer with a graphics card, or raise <b>Wait for the model up to</b> . For a cloud service, check that the server can reach the internet on port 443. |

## What is sent

- **Way 1:** what the AI reads (only what you ticked) goes to that AI app. If the app runs in the cloud, it goes to its provider.

- **Way 2:** nothing leaves your network.
- **Way 3:** for each question, the question and the facts Uplivra found go to the provider over the internet: measurements, alerts, and log message text if you allow it. **Device names, site names and IP addresses are replaced by stand-ins** before sending, unless an administrator chooses to send them as they are. Passwords and keys in log text are masked either way.

Every call is in the **audit log (Settings > Audit log)**: `mcp.call` for an AI app reading Uplivra (way 1), `ai.call` for each request to your model server or cloud service (ways 2 and 3).

## Turning it off

- **Way 1:** in **Settings > Connect an AI**, untick **Allow other AIs to read Uplivra** and click **Save**. Every connection stops at once.
- **Ways 2 and 3:** in **Settings > Connect an AI**, untick **Use an AI model** and click **Save**. To also remove a saved API key, tick **Remove the saved key**.

Insights keeps working without any AI: answers are then written by Uplivra's own analysis.