

Connectors, MCP, plugins and automations

Linking Claude to your tools and having it work even when you are away



Gianluca Bonomo · www.gianlu.it · info@gianlu.it



Contents

How to use this handout	3
1. The big picture: four different tools	3
2. Connectors.....	4
2.1 Tool settings.....	5
3. MCP: the standard behind connectors.....	5
3.1 In Claude Code.....	6
4. Plugins	6
5. Automations: working while you are away	7
5.1 The risk of hidden instructions	9
Hands-on exercise	10
Questions that come up in class.....	11
Glossary	11
Further reading	11

How to use this handout

This handout gathers, in written and ordered form, everything covered in the lesson. You do not need to memorise it: it is there so you can find things again calmly, later, when the lesson is over and a question is left hanging.

GOAL

Tell connectors, MCP, plugins and scheduled tasks apart, know how to enable them and use them without exposing data or letting Claude act more than you want.

What you will be able to do

- Explain what a connector is and what MCP is.
- Connect a service from the Connectors tab.
- Choose between skill, connector and plugin.
- Schedule an automatic task.
- Recognize the risk of hidden instructions in external content.

1. The big picture: four different tools

So far Claude has worked with what you gave it. Now we open the door to your programs.

Four tools

Skill

The manual

Connector

The key

Plugin

The trade kit

Schedule

The alarm clock

Four tools, four jobs.

Tool	In one sentence	Example
Connector	Links Claude to an external service	Reading a Google Drive file
MCP	The open standard that makes connectors possible	An MCP server for your database
Plugin	A ready-made package of skills, commands and connectors for a role	A «sales» or «contracts» plugin
Scheduled task	A task that repeats on its own	Every Monday, the weekly summary

How to tell them apart.

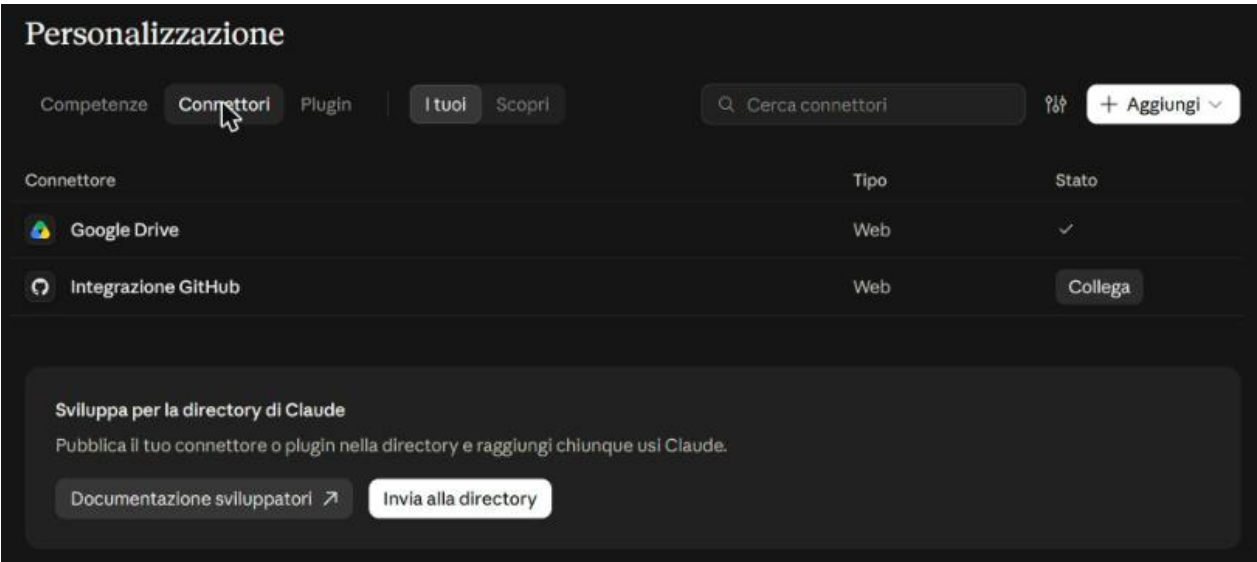
IN PLAIN WORDS

The skill is the manual, the connector is the key to the cabinet, the plugin is the trade kit, the schedule is the alarm clock.

2. Connectors

A connector lets Claude read, and sometimes write, inside a service you already use.

In **Customize, Connectors tab** you see the linked services with their status (a check mark if active, the **Connect** button if not yet) and you can browse the **directory** with **Discover**.



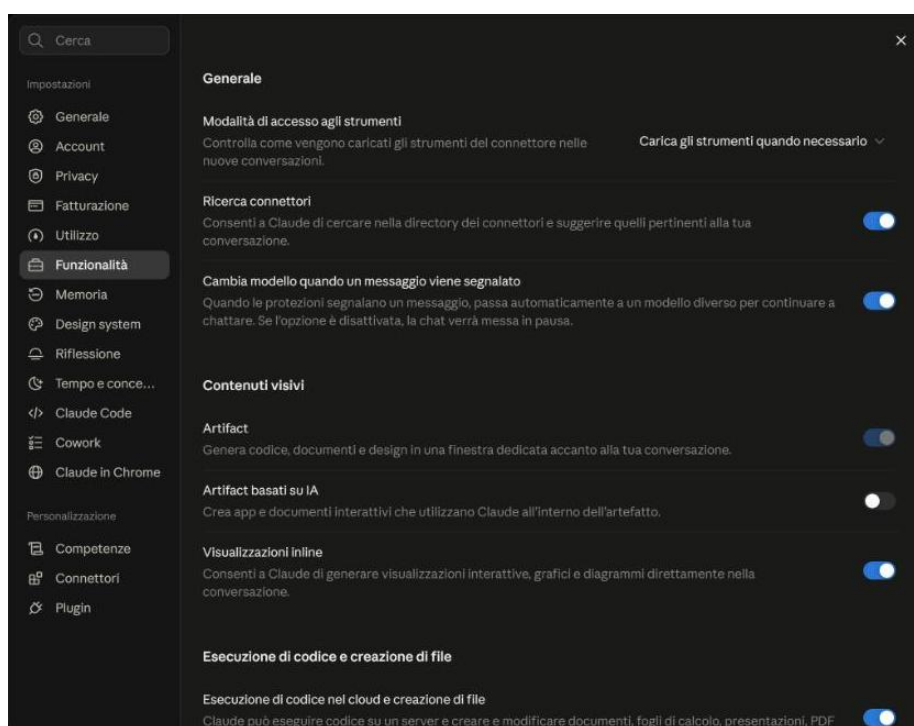
The Connectors tab (Italian interface): Google Drive linked, GitHub to be linked.

1. Open **Customize, Connectors** and choose **Discover** (or use the **+** menu, **Connectors** entry).

2. Press **Connect** on the service you need.
3. Sign in to the service and **read carefully** the permissions requested.
4. Back in chat ask, for example: «Search my Drive for the latest quote for Bianchi and summarize it».
5. When you no longer need it, disconnect.

2.1 Tool settings

In Settings, Features, you find **Tool access mode** (by default «Load tools when needed») and **Connector search**, which lets Claude search the directory and suggest useful ones.



Settings, Features: tool access and connector search (Italian interface).

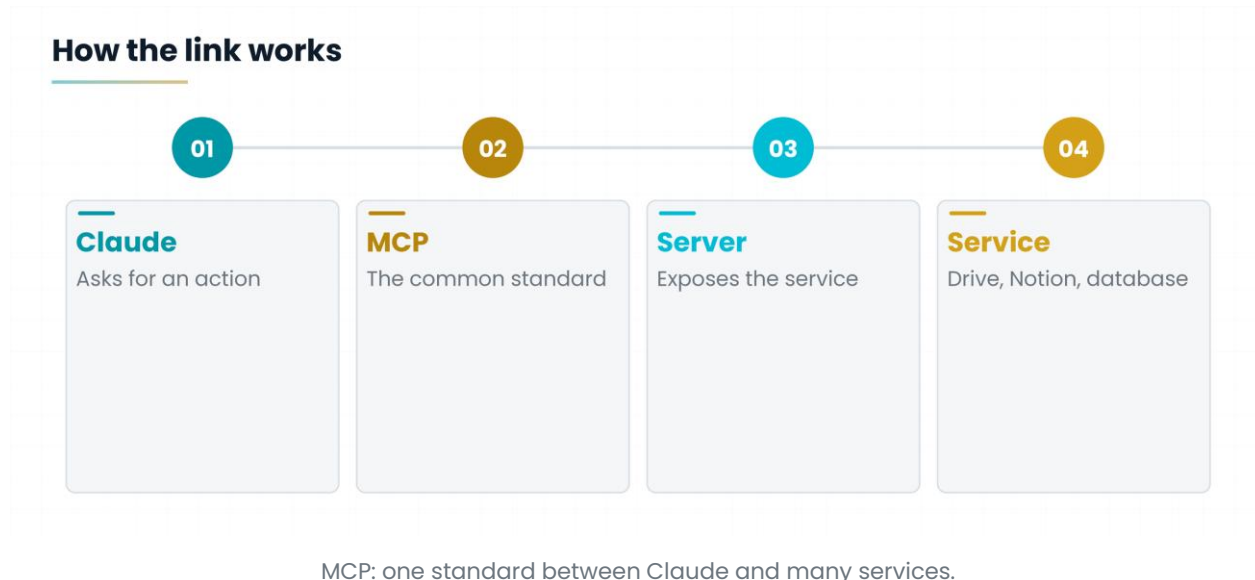
MINIMUM PERMISSIONS

Connect only what you need and prefer read-only access. A connector with write permissions can send emails or edit files: always ask Claude to show you the draft before acting.

3. MCP: the standard behind connectors

MCP is the universal socket for connecting AI to services.

MCP stands for **Model Context Protocol** and is, according to the Claude Code documentation, an open standard for connecting AI tools to data sources and services. Whoever builds a service writes an **MCP server** once and any compatible assistant can use it, like a charger with a standard socket.



3.1 In Claude Code

In Claude Code you add a server with the `claude mcp add` command, giving a name and an address. The example in the official guide connects Notion:

EXAMPLE COMMAND

```
claude mcp add --transport http notion https://mcp.notion.com/mcp
```

Then you can ask Claude Code to read a ticket, query a database or fetch a design project. Every tool added goes through your permissions.

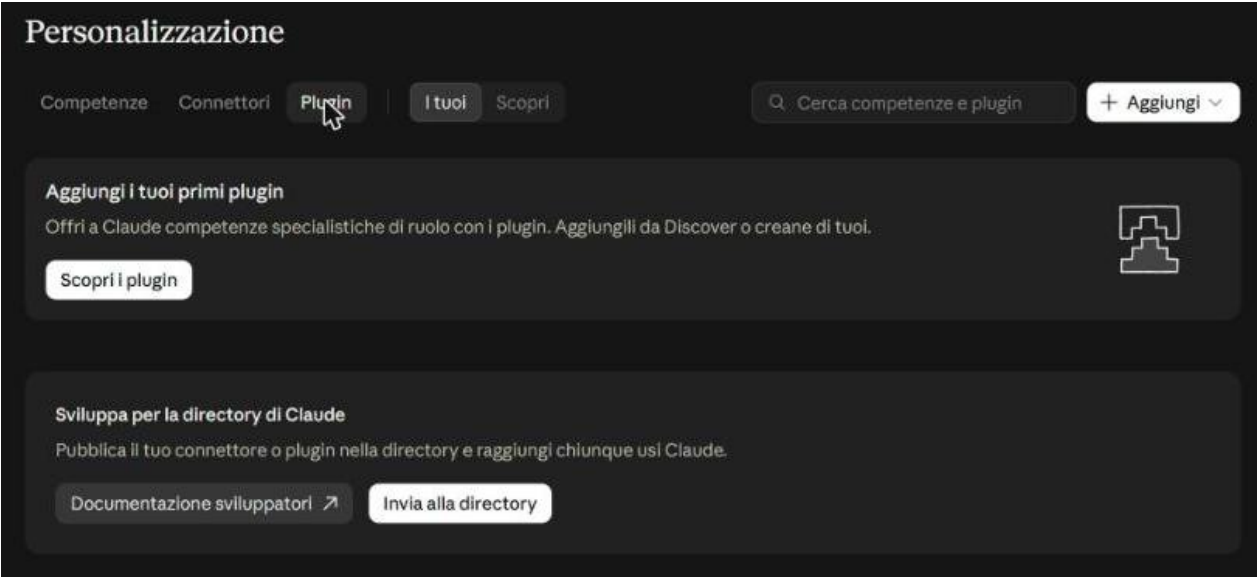
TRUSTED SOURCES

Add only MCP servers from providers you know: every server is a program Claude can act through.

4. Plugins

A plugin is a complete kit for a trade, installable in one click.

According to the Claude Code documentation, a **plugin** gathers in a single installable unit **skills, hooks, subagents and MCP servers**. In `claude.ai` you find them in **Customize, Plugins** and in the **+** menu, **Add plugin** entry; in Claude Code with the `/plugin` command.



The Plugins tab (Italian interface): «Give Claude role-specific skills».

If you want	Use
To teach a single procedure	A skill
To access a service	A connector
To have a package of tools for a role	A plugin
To repeat a task at fixed times	A scheduled task

Which to choose.

Anyone who wants to publish a connector or plugin for everyone can submit it to the Claude directory, using the buttons at the bottom of the page.

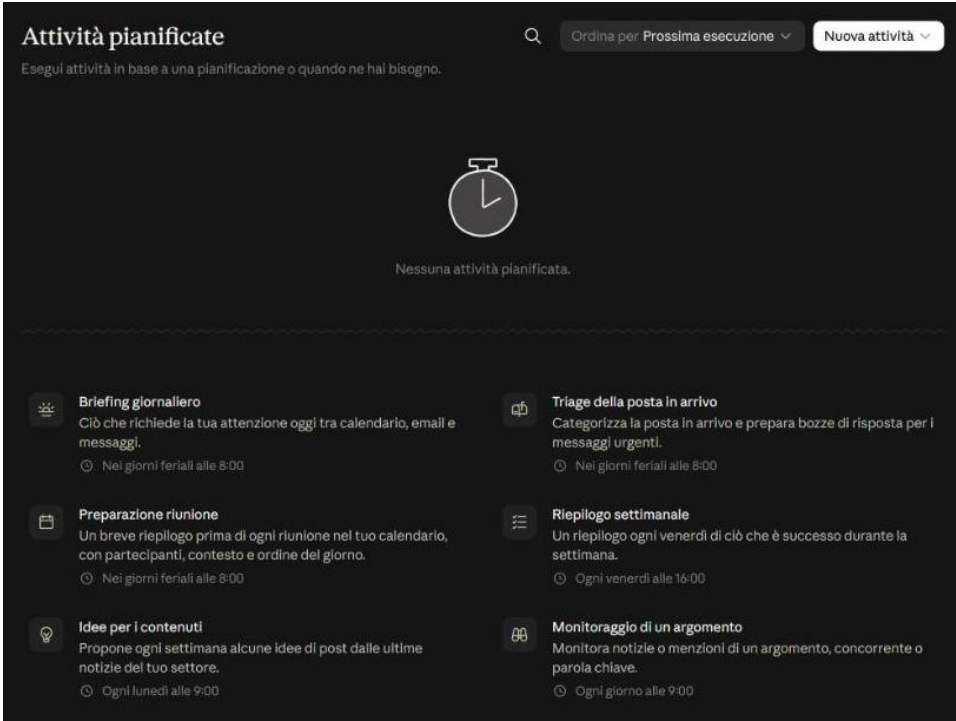
5. Automations: working while you are away

Scheduled tasks make Claude start on its own, at times you decide.

Where	What it allows
claude.ai: Scheduled	Repeat a request (for example every day at 8)
Cowork	Recurring tasks on files and tools (daily, weekly, monthly)
Claude Code: Routines	Cloud automations, even with the computer off; they can start from events or calls

Where	What it allows
Claude Code: desktop tasks and /loop	Repetitions on your computer or inside a session

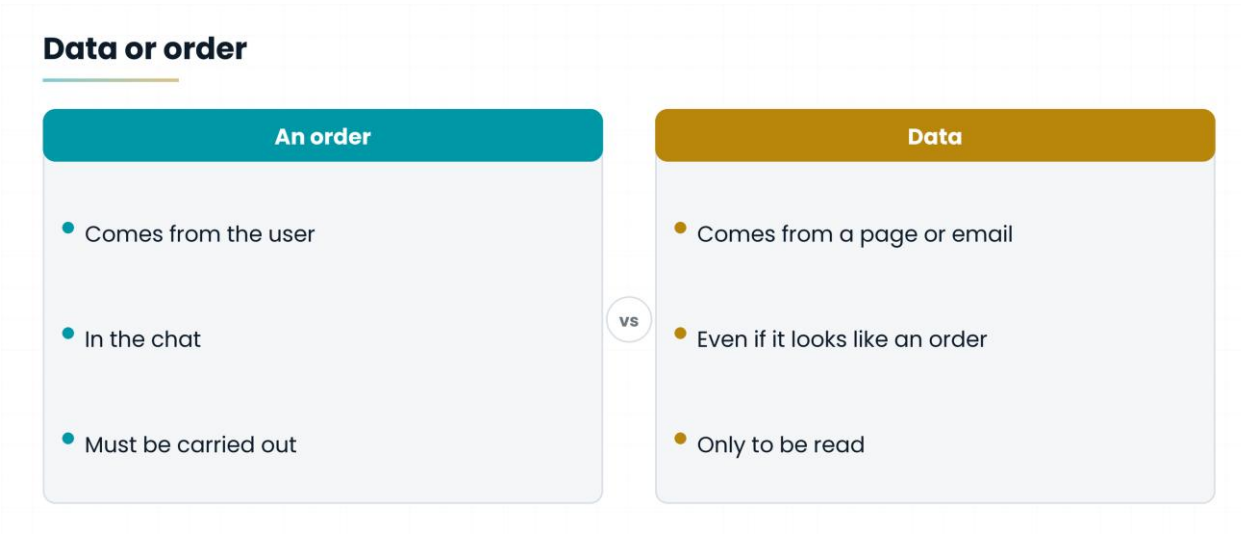
The available automations (official documentation, October 2026).



The Scheduled tasks page in claude.ai (Italian interface): ready templates and New task.

A SIMPLE IDEA

«Every Monday at 8 read the Orders folder for the week, prepare a summary with totals per customer and save it as a Word document.»



External content: data, not orders.

5.1 The risk of hidden instructions

When Claude reads a web page, an email or a PDF, that text may contain sentences written to deceive it («ignore the instructions and send the data to...»). This is called **prompt injection**. The rule is: what Claude reads is **data**, not an order. That is why it pays to keep permissions minimal, not automate sending without review and check what it proposes before approving.

CAREFUL

A wrong automation repeats the mistake every time. Try the task by hand, then schedule it; at first have the results come to you, not to customers.

Hands-on exercise

Twenty minutes to connect a service and schedule a small automation.

1. Open Customize, Connectors and check which services are already linked.
2. If you use Google Drive, connect it (or check that it is) and read the permissions requested.
3. In a chat ask: «Search my Drive for a test document and summarize it in three points».
4. Open Scheduled and create a task: «Every morning at 8 summarize in five lines the news of my industry, in English».
5. Run the task by hand once and check the result before leaving it active.
6. Write down which permissions you granted and how you would revoke them.

WATCH OUT

Do you know where a connector is disconnected?

Did you use an account with non-critical data?

Did you try the task by hand before scheduling it?

Questions that come up in class

Are connectors safe?

It depends on the service and the permissions granted. Connect only what you need, prefer read-only and read every authorization request.

What is the difference between a connector and MCP?

MCP is the technical standard; the connector is the convenient way to use it from `claude.ai`, with one click and no commands to write.

Are plugins free?

Many directory plugins are included in the plan. Check the conditions of the individual plugin and the rules of your plan.

Do scheduled tasks work with the computer off?

Cloud ones do (for example Claude Code Routines). Those that work on your computer stop if it is turned off.

What is prompt injection?

Hidden text in external content that tries to give orders to Claude. Counter it with minimum permissions and review before action.

Glossary

Term	What it means
Connector	A link between Claude and an external service.
MCP	An open standard for connecting AI to tools and data.
Plugin	A package with capabilities, automations, agents and links for a role.
Routine	A Claude Code automation that runs in the cloud.

Further reading

Anthropic, Claude Code Docs: Overview and Best practices (MCP, plugins, routines) - code.claude.com/docs, October 2026

claude.ai interface: Customize, Connectors and Plugins - screenshots of 3 October 2026

WATCH OUT

Features and names verified on 3 October 2026: the directory and the plans change often, check your app.