

AI Advances · [Follow publication](#)

🌟 Member-only story

How I Build MCP Servers and Host Them For Free

Wrap any Python function in an MCP in under a minute.

5 min read · May 9, 2025



Thuwarakesh Murallie

Follow



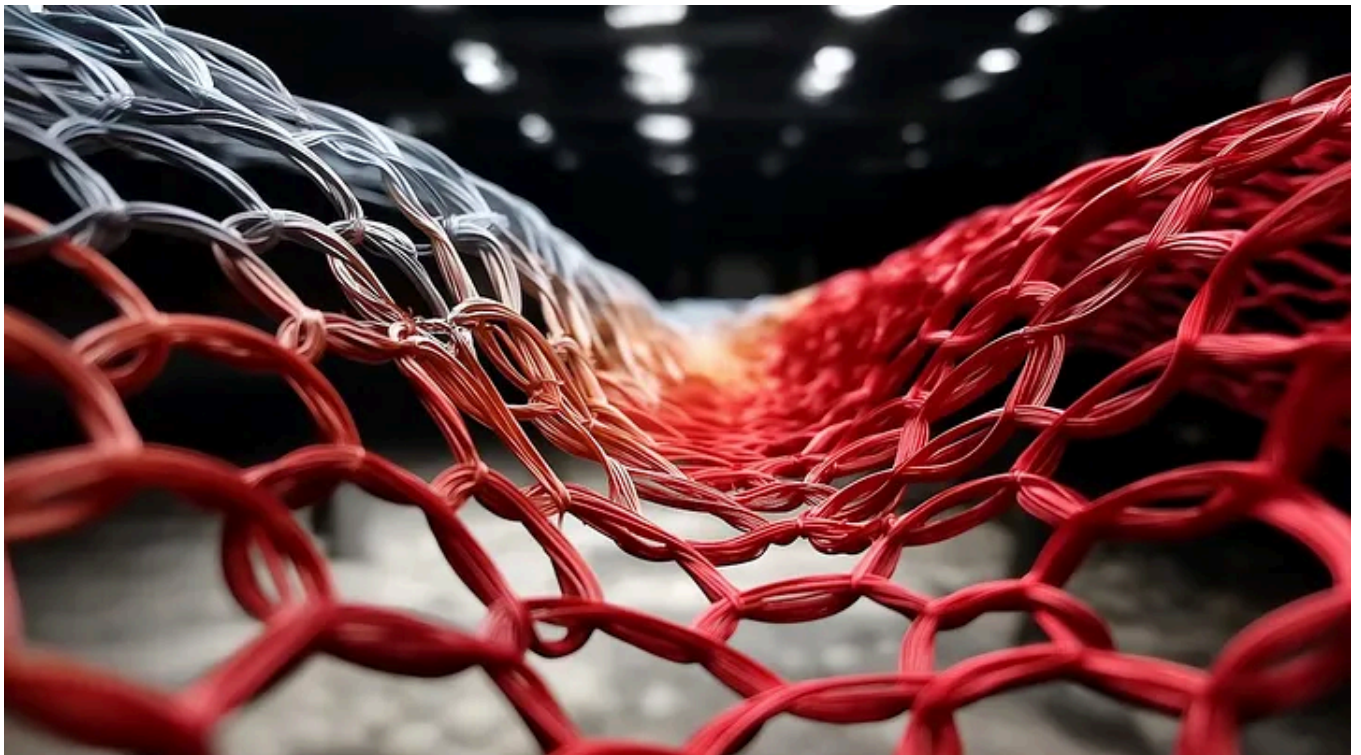
Listen



Share



More



[Image from Lummi.ai](#)

My first impression of MCP was, wow!

It's hardly anything different for others. We were all amazed when Antropic released this awesome protocol. Suddenly, a whole ecosystem was created overnight, with hundreds of MCP servers.

One thing remained a burning issue for me.

How could I create an MCP server for anything I want? How can I convert Python functions into MCP? How can I convert my private API endpoints into MCP?

The answer came to me only recently.

While skimming through my email newsletters, I saw a link to a [Huggingface](#). It says you can create an MCP with less than five lines of Python code. This is awesome.

I tried it, and now I've converted a ton of MCP servers for personal use.

Some of this can be achieved with existing MCP solutions, but they don't offer the granular control I needed. I get it with my custom MCPs, though.

Here's how it works.

Use Gradio Interface to convert Python functions into MCPs.

Gradio made it possible.

I knew Gradio was only a web interface builder. It still is, but now Gradio can do much more. We can build MCP servers with it.

I love Gradio because it functions as both a web app and an embedded tool within a notebook.

6 Python GUI Frameworks to Create Desktop, Web, and Even Mobile Apps.

You could build beautiful apps purely in Python.

medium.com



Here's how to build an MCP server. We'll also connect it to the Claude desktop. But first, let's start with the installation.

```
pip install "gradio[mcp]"

# or: uv add "gradio[mcp]"
```

Now, let's say we have a function that computes the distance of a projectile. It can be anything. But this one should work as an example.

```
def compute_projectile_distance(initial_speed, angle):
    """
    Calculate the horizontal distance traveled by a projectile.

    Parameters:
    initial_speed (float): Initial speed of the projectile in m/s
    angle (float): Launch angle in degrees

    Returns:
    float: Horizontal distance traveled in meters
    """
    # Convert angle from degrees to radians
    angle_rad = math.radians(angle)

    # Gravitational acceleration (m/s²)
    g = 9.81

    # Calculate the horizontal distance using the projectile motion formula:
    # distance = (initial_speed² * sin(2*angle)) / g
    distance = (initial_speed**2 * math.sin(2 * angle_rad)) / g
```

Now, we can make this an MCP. Here's how simple it is.

```
app = gr.Interface(
    fn=compute_projectile_distance,
    inputs=["number", "number"],
    outputs="number",
    title="Compute Projectile Distance",
    description="Computes the theoretical horizontal distance a projectile trav
)
```

With this one API call, our app can serve as an MCP. However, the following line creates the actual MCP server.

```
app.launch(mcp_server=True)
```

We can now start the app as usual.

```
python app.py
```

This would start our MCP server at http://localhost:7860/gradio_api/mcp/sse.

This is now ready to be configured for any client. Claude Desktop, Cursor, and Cline are popular MCP clients. You can also build your client too.

In this post, I'll show you how to integrate it with Claude Desktop.

Along with the MCP server, Gradio also starts the app's web interface. You can access this at <http://localhost:7860>. At the bottom of the web interface, you'll see a button called "Use via API."

Click on it.

Screenshot by the author.

It opens a popup window. There, you see a button called MCP. Click on it to see instructions on how to set up your MCP server with clients.

Screenshot by the author

I use the second method highlighted here, which uses Node.js. For this to work, you should have Node.js installed on your computer.

Copy this and paste it into the config file of Claude Desktop. You may find it in the installation location with the name, `claude_desktop_config.json` .

But if you can't find the installation location, do this.

Go to Claude Desktop and click on the hamburger menu at the very top left corner of the app. Then go to file and click settings. Finally, go to developer settings and click on the edit config button.

Once you've pasted the configuration, you have to restart Claude Desktop. And there you go. Claude Desktop can now call your function to compute projectile distances.

Let's try it.

Screenshot by the author.

In the above prompt, I ask if Tim throws a stone at 2m/s velocity and a 30-degree angle, will it hit Sam? Sam is 25 meters away from Tim.

Claude didn't do the calculation by itself. Instead, Claude used the tool we provided (through MCP). Based on the tool's result, the model concluded that the stone wouldn't hit Sam.

See! Creation of an MCP, Starting the MCP server, connecting with an MCP client, and using the tools all came effortlessly. That's why I love this method.

MCP created; how about hosting?

Private MCPs need hosting. You need to handle it.

But you know what? Put it in a HuggingFace Space, and you will get free hosting.

If this is the first time you publish on Spaces, it works a lot like Git.

Go to HuggingFace Spaces and click on “New Space.” Fill in the details and select Gradio for Space SDK. Leave the template blank.

When you first create a Space, you get the instructions to connect the Space with your local codebase. If you've ever worked on Git, forgive me for boring you.

Make sure to add a `share=True` to the `app.launch` method.

```
app.launch(mcp_server=True, share=True)
```

Without this, your app and the APIs are available only in the same environment. But when we host it on the cloud, we expect to access it from our local computer. When you set this True, Gradio will give you a public URL, which you can now replace in your client (Claude Desktop)

Final thoughts

MCP is new, so I can't claim I'm an expert.

I knew how to connect MCP servers to Claude Desktop and Cursor but didn't know how to create one. I thought it would be like a PhD thesis.

But Gradio made it very easy.

I've created dozens of MCPs for my personal use now. Have you created MCP servers? If not with Gradio, what's your preferred method?

I'd love to hear them.

. . .

Thanks for reading, friend! Besides Medium, I'm on LinkedIn and X, too!

Python

Genai

Mcp Server

Llm

AI



Follow

Published in AI Advances

53K followers · Last published just now

Democratizing access to artificial intelligence



Follow



Written by Thuwarakesh Murallie

5.6K followers · 149 following

Lead Data Engineer @ Stax, LLC.

Responses (18)



Bgerby

What are your thoughts?



Remy Schmilinsky he/him

May 11



Thanks for sharing. Here's another approach for hosting MCP-compatible inference services at no cost and no dependency on any SaaS whatsoever: <https://github.com/schmitech/orbit>



26

Reply



mohamad shakhajeh

May 13



Interesting approach! It's great to see how you've leveraged Gradio to simplify the process of turning Python functions into MCPs. I like how you emphasized the control that custom MCPs offer. The potential for personal use is definitely exciting!



21

[Reply](#)



Varun Narula

May 14



Here's an approach where the python code does not exist, an entirely new MCP server needs to be created.

The author uses VSCode, Roo Code, MCP docs to do this, take a look ... [more](#)



5

[Reply](#)

See all responses

More from Thuwarakesh Murallie and AI Advances

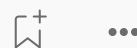



In Level Up Coding by Thuwarakesh Murallie

I Tried 20+ Open-Source LLMs. These 5 Perform Remarkably Well on CPUs.

If you can't afford a GPU but need to handle inferences privately, this is the solution.

★ May 5 🤝 689 💬 13

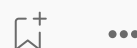



 In AI Advances by Nikhil Anand

I wasted months running slow LLMs before learning this

Why your LLM is running at just 10% of its potential speed

★ Oct 10 🤝 910 💬 11




 In AI Advances by Kuriko Iwai

Mastering Autoencoders (AEs) for Advanced Unsupervised Learning

Explore the core mechanics of AEs with essential regularization techniques and various layer architectures

★ Oct 2 🖱️ 457 💬 3



 In AI Advances by Thuwarakesh Murallie

This NumPy Alternative is built for GPUs and is 25X faster.

And it needs almost no changes to your NumPy code.


★ Apr 11 🖱️ 424 💬 6



See all from Thuwarakesh Murallie

See all from AI Advances

Recommended from Medium


 In Towards AI by Teja Kusireddy

We Spent \$47,000 Running AI Agents in Production. Here's What Nobody Tells You About A2A and MCP.

Multi-agent systems are the future. Agent-to-Agent (A2A) communication and Anthropic's Model Context Protocol (MCP) are revolutionary. But...

Oct 16  1.8K  49



 Nivetha Thangaraj

Kubernetes Is Dead: Why Tech Giants Are Secretly Moving to These 5 Orchestration Alternatives

By Nivetha Thangaraj

Jun 30  777  101



 Tosny

7 Websites I Visit Every Day in 2025

If there is one thing I am addicted to, besides coffee, it is the internet.

★ Sep 23  6.3K  227



Claude Desktop Might Be the Most Useful Free Tool You'll Install This Year

I didn't expect much when I first saw the announcement for Claude Desktop. Another AI wrapper, I thought. Maybe with a shiny UI.

★ Oct 23 🖱️ 382 💬 15




○ Abhinav

Docker Is Dead—And It's About Time

Docker changed the game when it launched in 2013, making containers accessible and turning “Dockerize it” into a developer catchphrase.

★ Jun 8 🖱️ 7.2K 💬 204



 In AI Software Engineer by Joe Njenga

This Viral DeepSeek OCR Model Is Changing How LLMs Work

This DeepSeek OCR model hit an overnight success not seen in any other release—4k+ GitHub stars in less than 24 hours and more than 100k...

✦ Oct 23 🖱️ 294 💬 4



See more recommendations