Attaching Claude Code To MCP

Having your Claude Desktop app set up with MCP and connected to Docker with clients means you're about 80-90% of the way there. But Claude Code (the Claude IDE/agent inside Cursor or Claude Desktop) needs a few specific things to recognize and interact with your local MCP agents as development tools (like invoking agents from prompt files, running server-backed logic, etc.).

Here's a **step-by-step breakdown** of what's really happening — and what's left to connect Claude Code to your MCP setup via Docker.



Step 1: Understanding the Stack

You Already Have:

- Claude Desktop installed
- Docker running
- MCP server container set up and connected
- Claude can talk to mcp_server.py or agents/ locally

You Might Still Need to:

- Explicitly register Claude Code agents inside your MCP folder
- Link Claude Code to the local repo you're editing in Cursor
- Mount your Cursor project into Docker so Claude can see .md agent files
- Set proper .env or volume bindings so Claude Code and MCP share context

X Step-by-Step Setup to Connect Claude Code to Your MCP via Docker



1. Create a Shared Volume for Your Project

Ensure your docker-compose.yml (or docker run) mounts your local codebase (e.g., ~/projects/16bitfit) into the container.

Example:

yaml

CopyEdit

volumes:

- ./your-local-project:/app

If you're not using docker-compose, this would be something like:

bash

CopyEdit

docker run -v ~/projects/16bitfit:/app my-claude-container

This makes your agents/ folder and code visible to Claude Code inside Docker.

2. Make Sure Your Claude MCP Agent Folder Is Set

Inside your Claude MCP repo, you should have something like: bash

```
CopyEdit
~/claude-mcp/
  L---- clients/
     16bitfit/
              - agents/
              mcp_server.py
              - .env
              - app.py
Update your mcp_server.py to look for agents in /app/agents (where you
mounted your code in Step 1).
In mcp_server.py:
python
CopyEdit
import os
AGENT_DIRECTORY = os.getenv("AGENT_DIR", "/app/agents")
In .env:
dotenv
CopyEdit
AGENT_DIR=/app/agents
```

▼ 3. Tell Claude Desktop Where the MCP Client Lives

Claude Desktop typically reads from .claude/ or asks for a config file pointing to a registered MCP client. Make sure your MCP config looks something like this: ison

```
CopyEdit

{
    "client": "16bitfit",
    "description": "Phaser 2D Fighting Game App",
    "path": "/Users/you/claude-mcp/clients/16bitfit"
}
This file should as incide:
```

This file should go inside:

bash

CopyEdit

~/.claude/clients.json

You may have already done this if Claude Desktop is showing your connected agents.

4. Link Your Cursor Project to the Same MCP Client Folder

Make sure the project you're working on in **Cursor** is also using /Users/you/claude-mcp/clients/16bitfit or has symbolic links to it:

bash

CopyEdit

In -s ~/projects/16bitfit ~/claude-mcp/clients/16bitfit

Or better yet, keep your full project there if that's where Docker and Claude are referencing files.

▼ 5. Restart Claude Desktop and MCP Container

Once all paths are mounted correctly and the .env is updated, restart everything: bash

CopyEdit

docker restart < container-name>

Then quit and reopen Claude Desktop.

6. In Cursor or Claude Desktop, Run a Test Agent Prompt

Try something like:

"Use the PhaserFighterAgent from /app/agents/phaser-fighter-agent.md to create a horizontal fighting scene."

Claude should:

- Recognize the .md file
- Load the prompt context from Docker volume
- Use that personality for your Claude Code session

Optional Debug Tip

Inside the container, run:

bash

CopyEdit

Is /app/agents

cat /app/agents/phaser-fighter-agent.md

This confirms that:

- Your agents/ folder is mounted
- Claude (via mcp_server.py) can access it

Summary

 ▼ Task	Status
Docker running	\checkmark
MCP server configured	\checkmark
Project mounted to Docker	
.env points to /app/agents	
Claude Desktop client points to 16bitfit path	
.md files placed in agents/ folder	\checkmark
Prompt test working in Claude Code	