MCP Server and client with Claude

Last amended: 16th Oct, 2025
Folder: C:\Users\ashok\OneDrive\Documents\claude
Reference this article

Connect to local MCP servers

> Learn how to extend Claude Desktop with local MCP servers to enable file system access and other powerful integrations
>
> Model Context Protocol (MCP) servers extend AI applications' capabilities by providing secure, controlled access to local resources and tools. Many clients support MCP, enabling diverse integration possibilities across different platforms and applications. This guide demonstrates how to connect to local MCP servers using Claude Desktop as an example, one of the many clients that support MCP. While we focus on Claude Desktop's implementation, the concepts apply broadly to other MCP-compatible clients. By the end of this tutorial, Claude will be able to interact with files on your computer, create new documents, organize folders, and search through your file system—all with your explicit permission for each action.

Step1: Create a free account in Claude. It is free forever with certain limitations of usage and prompt limits. For limitations, please read here.

Step2: Download *Claude Desktop* from here and install on your machine.

Step3: Reboot your machine.

Step4: Open *Claude Desktop*. Press ctr+alt+space to open a chatbot and ask a simple question.

Step5: Install *node.js*, open source java script. Download Windows msi file (LTS version) from here and install.

Step6: Open command prompt and enter the command `node -v`. You should get an output something like below:

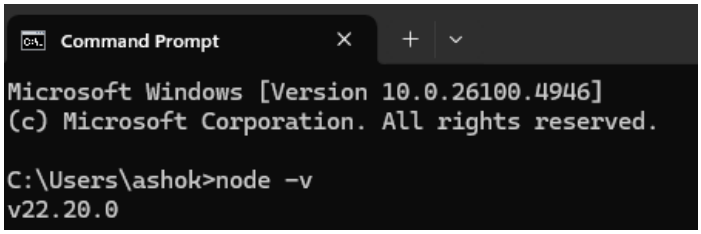

Figure 1: Checking node.js installation

Step7: Reboot your machine.

Step8: Open Claude Desktop and access Settings, as:

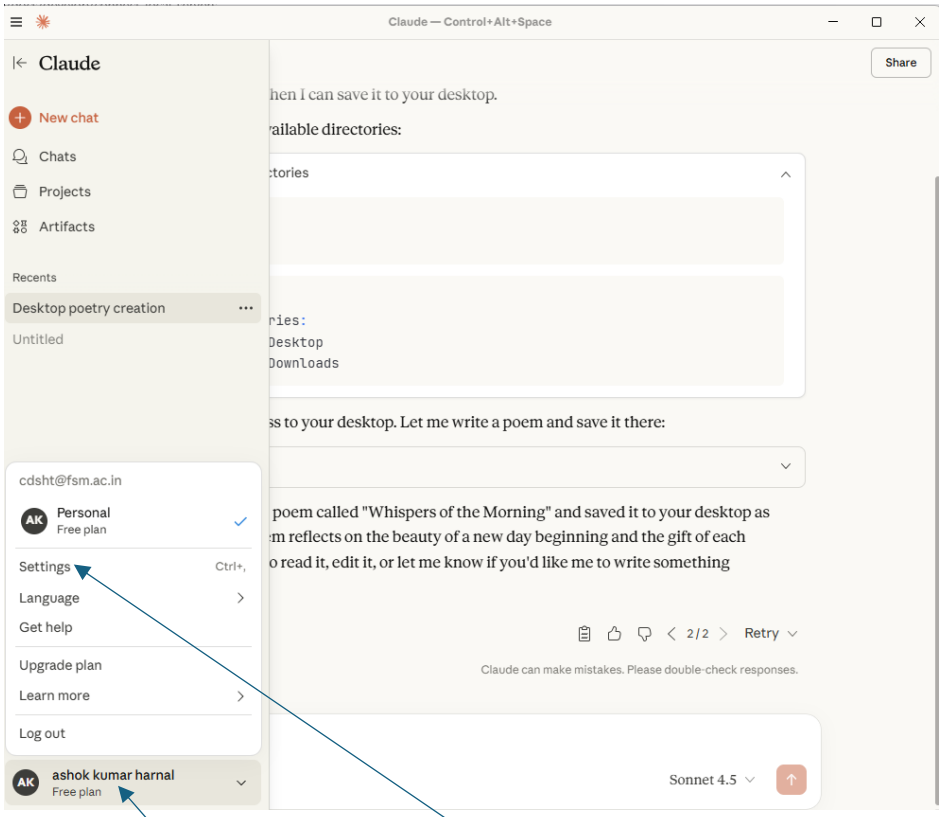

Figure 2: Click here to open menu to access Settings.

Step9: In Settings window, click on Developer tab to open it.

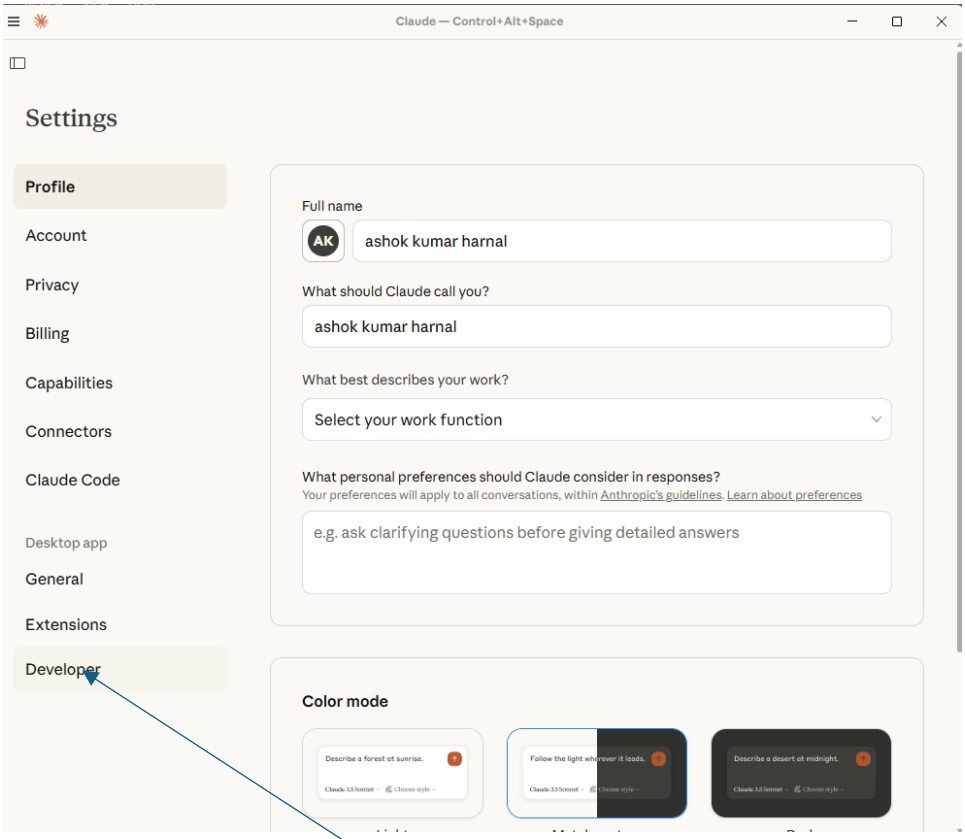

Figure 3: In Settings window, click Developer

Step10: In the Developers window, click on Edit Config. At this time, no local MCP server is installed.

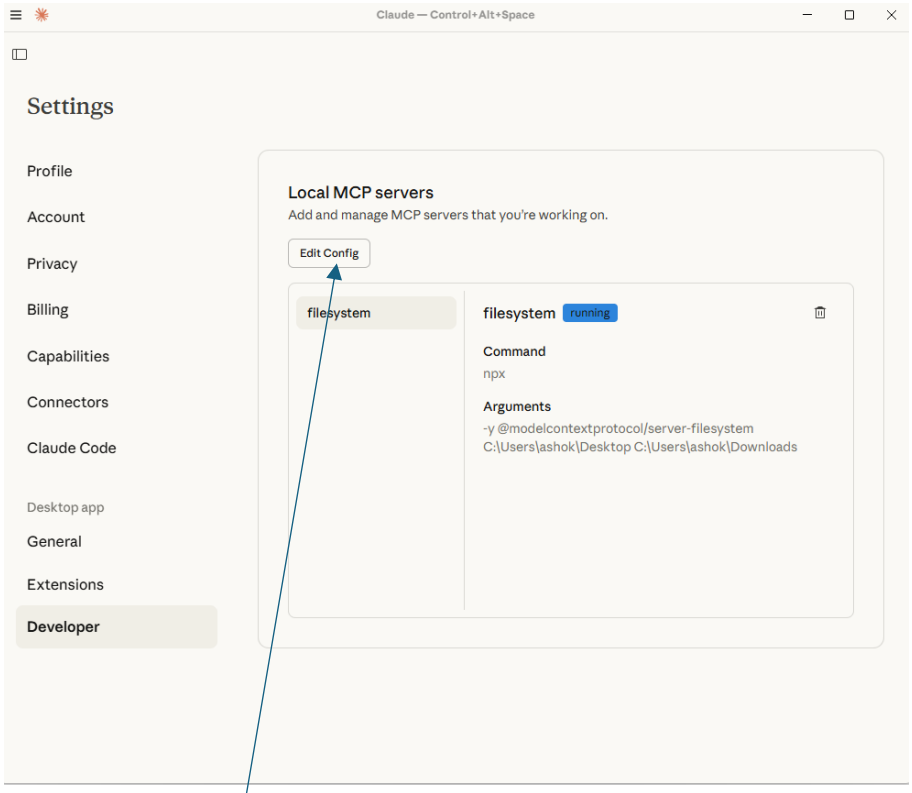

Figure 4: Click on Edit Config button

Step11: Clicking *Edit Config* takes you to a file: `claude_desktop_config.json`. Open this file in VSCode or notepad++, and replace its contents with:

```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "C:\\Users\\username\\Desktop",
        "C:\\Users\\username\\Downloads"
      ]
    }
  }
}
```

Replace username with the name of your machine, for example: ashok. This is the file on my machine:

```
{
  "mcpServers": {
    "filesystem": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-filesystem",
        "C:\\Users\\ashok\\Desktop",
        "C:\\Users\\ashok\\Downloads"
      ]
    }
  }
}
```

Save the amended json file and reboot the machine.

Step:12: Upon restart, open claude desktop, press ctrl+alt+space. You will see an icon to converse with MCP servers having tools:

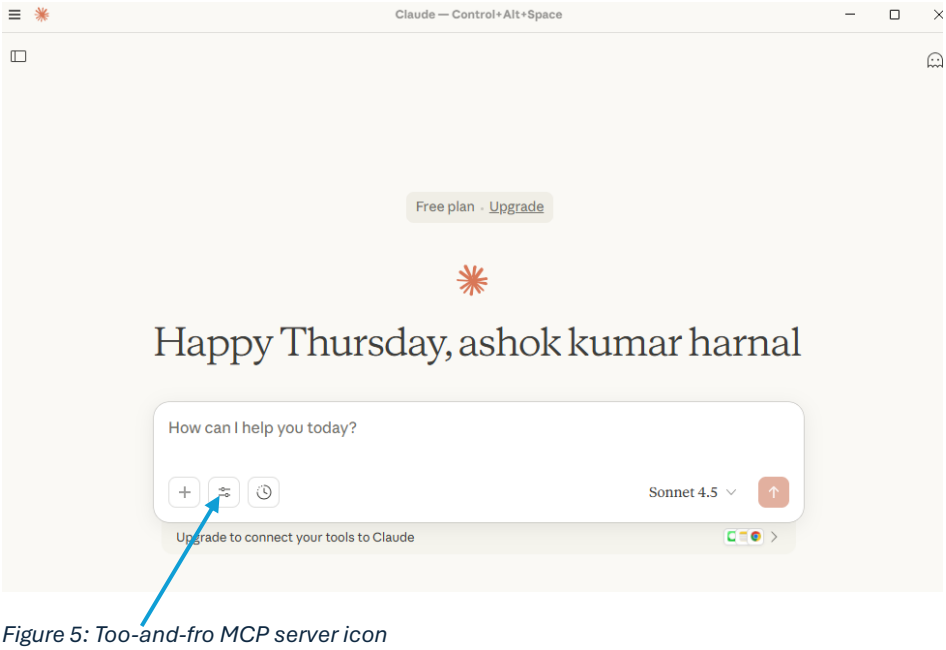

Figure 5: Too-and-fro MCP server icon

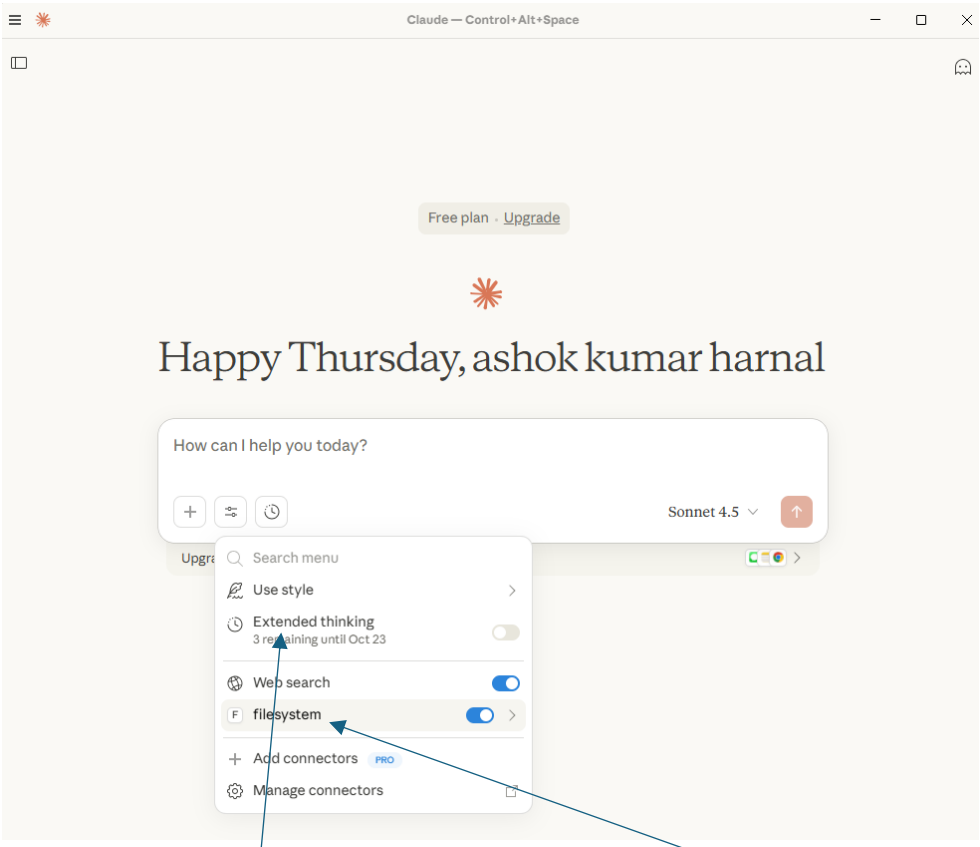

Figure 6: Click on the MCP server icon. Among tools, you will find filesystem tool.

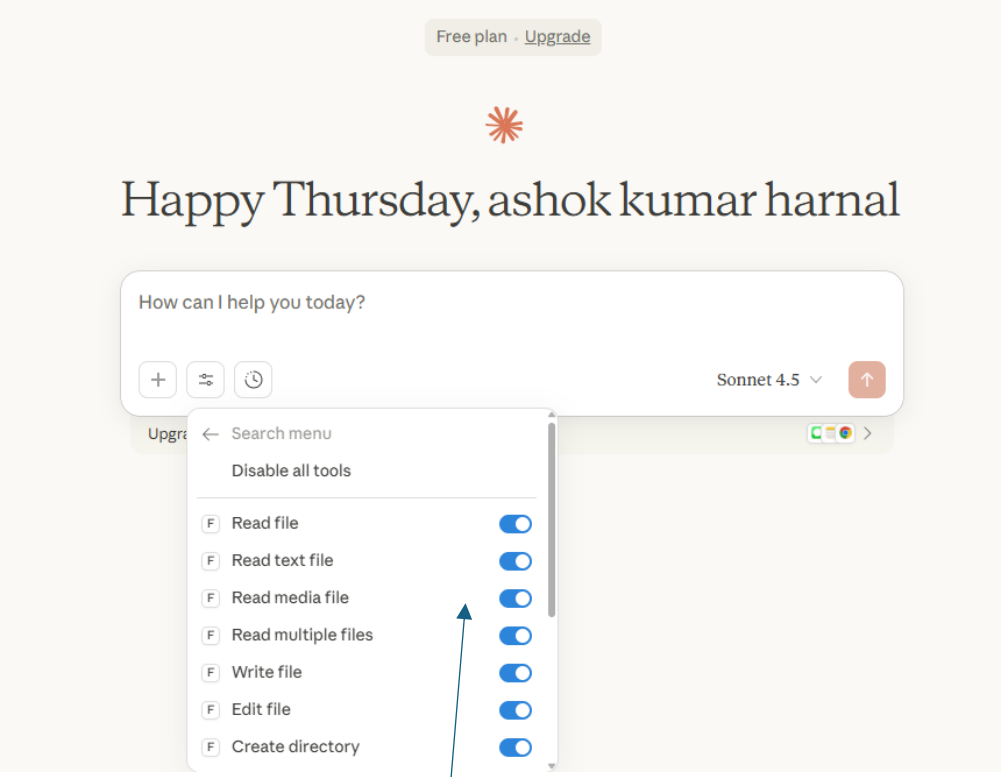

Figure 7: Click on filesystem to see what all agent tools are available. We have many.

Step13: Begin Experimentation:

With the Filesystem Server connected, Claude can now interact with your file system. Try these example requests to explore the capabilities:

File Management Examples

- **"Can you write a poem and save it to my desktop?"** - Claude will compose a poem and create a new text file on your desktop
- **"What work-related files are in my downloads folder?"** - Claude will scan your downloads and identify work-related documents
- **"Please organize all images on my desktop into a new folder called 'Images'"** - Claude will create a folder and move image files into it

##########