MCP tools for your Homelab: Building LLM Sysadmins

Daniel Colson

LLM Development Speed Reminder

- This information may be out-of-date next month 😜
- MCP is a relatively new standard
- New transport method (Streamable HTTP) and auth method (OAuth) were added to the MCP standard while I was writing this presentation.
- Major changes/improvements in the LLM ecosystem every few months.

- However, tool/function calls aren’t going anywhere, and support for MCP in particular is growing.

Topics Overview

- Model Context Protocol for providing tools and resources to LLMs
- General usage of MCP with LLMs
- Specific use cases for LLM agents using MCP tools in a homelab
- Related side topics when useful

Resources: https://github.com/Red5d/SELF2025

Not Covered: Prompt Engineering and local LLM setup

Model Context Protocol

MCP Overview

Developed by Anthropic (makers of Claude LLMs) in November 2024.

Standardized method for providing tools/functions for LLMs to understand and use.

Easy to get started.

Large (and growing) collection of existing MCP servers/functions available now.

Model Context Protocol

MCP Features and Capabilities

MCP servers provide clients with the ability to list and run/retrieve the following types of items/data:

Tools

- Functions for the LLM to choose from, fill in parameters and run

Resources

- Data that can be added into the LLM prompt to provide additional context

Prompt Templates

- Example prompts for the user to select from with optional parameters
- Can be used for things like slash (/) commands in LLM chat

MCP Server Connection Methods

STDIO (run script directly, standard IO)

- Local connection only
- Server is started by the calling program
- Actions execute on the local host

SSE (Server Sent Events) / Streamable HTTP

- Remote connection
- Security considerations
- Can run on remote host
- Actions execute on remote host

https://modelcontextprotocol.io/specification/2025-03-26/basic/transports

Security

The latest MCP specification (as of May 2025) includes the following authorization methods for HTTP-based transports:

- OAuth 2.1
  - Authorization Code: useful when the client is acting on behalf of a (human) end user
  - Client Credentials: the client is another application (not a human)
- Access Token
  - Token value in the “Authorization” request header field
  - Authorization: Bearer <access-token>

https://modelcontextprotocol.io/specification/2025-03-26/basic/authorization

Competing/Alternate Methods

Competing/Alternate Methods

OpenAPI Server

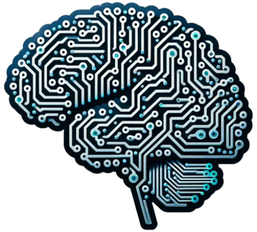
- Scalable, structured
- Less dynamic than MCP
- More initial structure required (depending on use case)

Tool Functions

- Integrated within application
- Less scalable
- Less dynamic

Agents and API Calls

Agentic vs Reasoning-mode



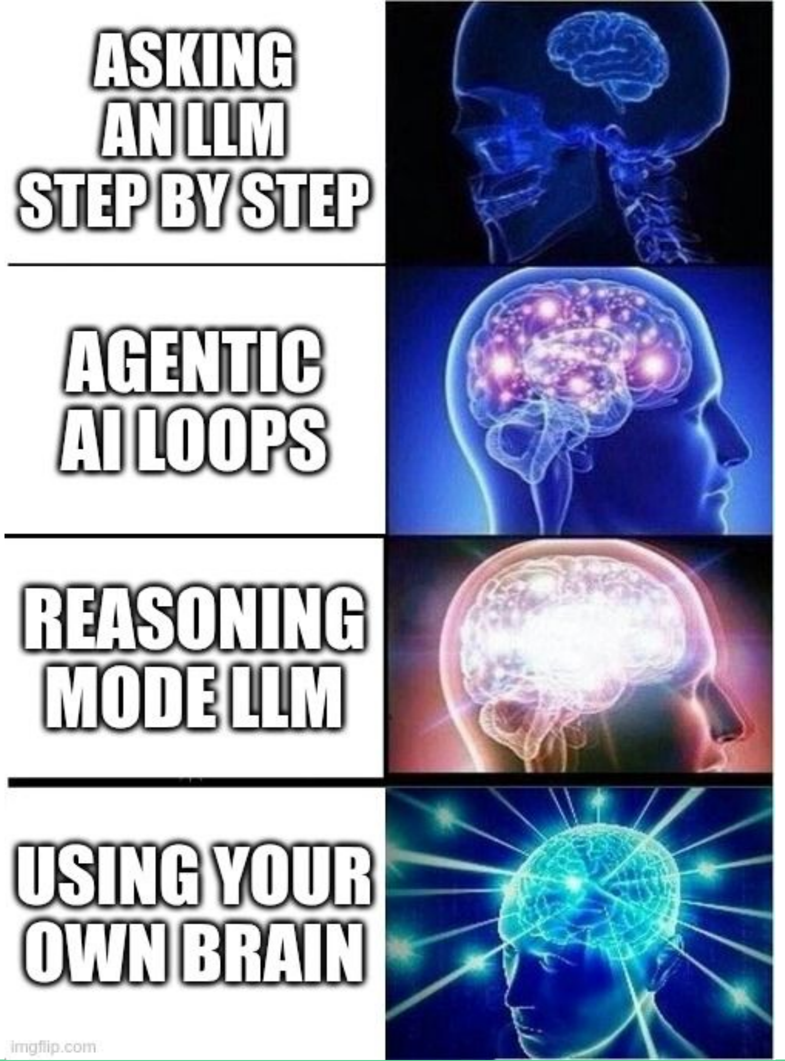
- Agents are multiple LLM calls in a loop. Planning and re-evaluating at each step.
- Takes actions as needed during each loop until the goal is achieved.

- Reasoning-mode LLMs “think” many times in a “loop” internally within a single LLM call before returning a result.
- Cannot take actions mid-thought.

Both can accomplish similar results, but Agents are more efficient at completing goals that require taking multiple actions and re-evaluating during the process.

Agentic vs Reasoning

Because of the ability to take actions during the “thinking” cycle, we’ll focus on Agents for purposes of this talk

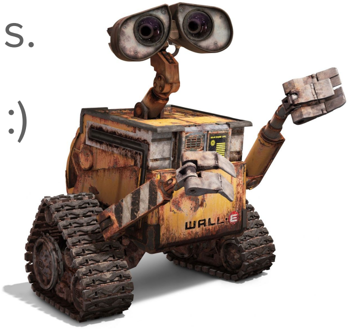


Agents vs Traditional Scripting

Do you need an LLM? Why not just call the API or run the function and process it?

Situations for an LLM:

- The returned results may be unpredictable, unstructured, or difficult to interpret directly in a programmatic way.
- You could easily provide a human with instructions on how to do a task, but it would be difficult or take a while to script and handle all conditions.
- You're ready for the robots to take over and do your work for you :)



Building MCP Functions

Considerations for Building LLM Functions

Input Validation (what happens if the LLM provides invalid input?)

Safe Actions (Could something potentially bad happen as a result of this function receiving a variety of inputs?)

Prompts and Descriptions (making sure the function is accurately described to the LLM so it knows how and when to use it)

Should this even <u>be</u> an agent-controlled function? (use structured outputs to get data back from LLM and process it with “normal” code?)

Auto-run vs user-confirmed actions?

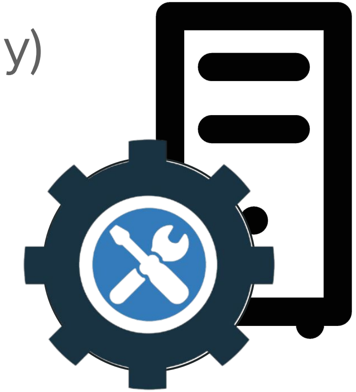
Sysadmin-Related Functions (for LLMs)

Actions that I’m ok with LLMs having the ability to run (with limitations):

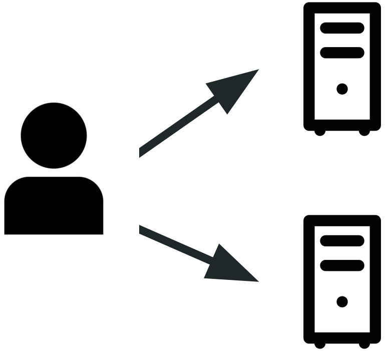
- Restarting containers/services
- Reading logs, config files, OS/system info, etc.
- HTTP/Ping/DNS checks
- Running package updates, installing packages (probably)
- More?

Not OK:

- Running arbitrary commands
- Directly managing files/storage



Homelab Agent Structures



Individual agents on each server

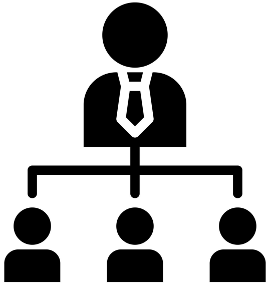
VS

A single agent with connections to MCP functions on each server via SSE

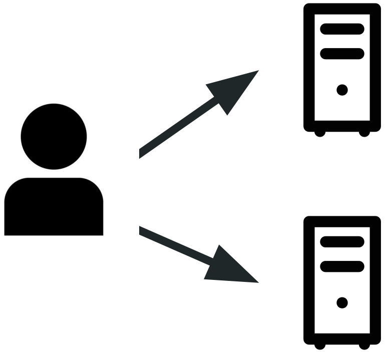
Single agent with all tools

VS

Managed team of scoped agents with tools for different tasks



Homelab Agent Structures

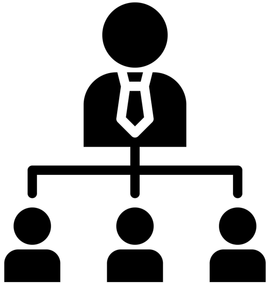


Individual agents on each server

VS

A single agent with connections to MCP functions on each server via SSE

Single agent with all tools



VS

Managed team of scoped agents with tools for different tasks

My Setup

The Parts

- **FastMCP** - https://github.com/jlowin/fastmcp
  - MCP Server/API development
- **mcpadapt** - https://github.com/grll/mcpadapt
  - Adapts MCP functions to the various “Tool” formats for different LLM frameworks
- **smolagents** - https://huggingface.co/docs/smolagents/index
  - Easy-to-use LLM Agent framework from Huggingface

Alternate agent frameworks: CrewAI, Langchain, LlamaIndex, Autogen

Sysadmin Agent Demo

Fixing a stopped Docker container service

https://github.com/Red5d/SELF2025

Personal MCP Demo

Info about me for your LLM

https://github.com/Red5d/me-mcp

Enterprise Uses

Agentic/MCP/Tool uses at $dayjob

Cloud and security vendors starting to offer MCP servers:

- AWS
- GitHub
- Cloudflare
- Crowdstrike
- Wiz
- ...

Can be used either for agentic automations or interactively for security analysts to dig into alerts, connect data points, and take actions.

Where to find MCP servers to run/use:

MCP Github: https://github.com/modelcontextprotocol/servers

Docker Hub - https://hub.docker.com/catalogs/mcp

Glama - https://glama.ai/mcp/servers

Smithery - https://smithery.ai/

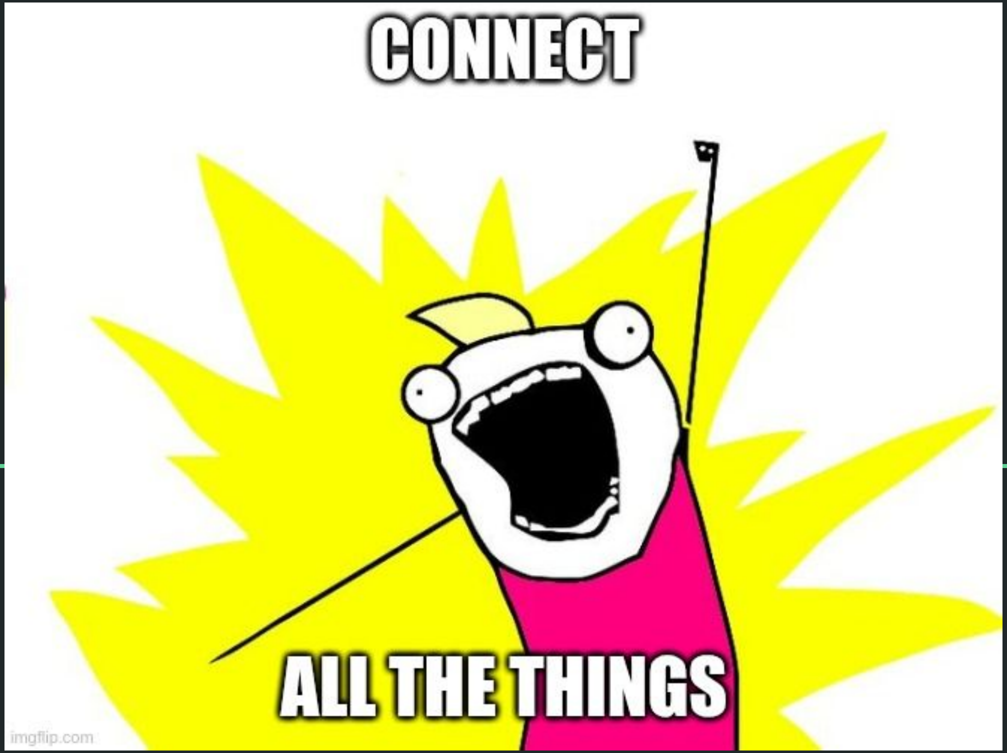
Pipedream - https://mcp.pipedream.com/

Awesome List - https://github.com/punkpeye/awesome-mcp-servers

A few interesting MCP servers

- **GitHub/GitLab/Gitea/etc...**
- **Use a web browser** - playwright-mcp
- **Analyze Windows Crash Dumps** - mcp-windbg
- **Accurate NixOS package and config option info** when writing NixOS configs - mcp-nixos
- Video editing through **ffmpeg** or **DaVinci Resolve**
- **SSH, Docker, Kubernetes, Terraform, CI/CD** tools
- MCP servers that **install/run other MCP servers** as needed
- **DoorDash** ordering - DoorDash-MCP-Server

The Future



On my systems...

- Add more sysadmin/networking functions to my MCP server(s)
- Set up the MCP server to run on the rest of my servers
- Send monitoring alerts for my services to an LLM agent workflow to investigate
- Add a way for the agent to contact me with the results
  - Matrix chat?
- Schedule the agent to check certain things regularly.

Other Uses

MCP standard (if adopted wide enough) offers the potential for LLMs to be able to “natively” discover and use APIs without having to write custom integrations for every service.

Other things that support MCP:

- Home Assistant (control devices, etc)
- Microsoft Windows (soon)
- Pipedream (https://mcp.pipedream.com/)

Interactive MCP

- AnythingLLM - **https://anythingllm.com/**
- Open WebUI (via their “mcpo” proxy)
- VSCode (and similar LLM-enabled code editors)
- Claude Desktop
- More here:
  - **https://www.pulsemcp.com/clients**
  - **https://github.com/punkpeye/awesome-mcp-clients**

Contact

Contact methods linked here: **https://red5d.dev**

GitHub: github.com/Red5d

Matrix: https://matrix.to/#/@red5d:red5d.dev

X: @red5_d

Also accessible via my informational MCP server:

https://mcp.red5d.dev/sse