

[Event dashboard](#) > Lab 3: MCP

Lab 3: MCP

Overview

In this lab, you will learn how to integrate Model Context Protocol (MCP) servers with Kiro to enhance your AI agent's capabilities. You'll configure MCP servers, access external documentation, and use specialized tools to improve your software development workflow. This lab demonstrates how MCP extends Kiro's functionality by connecting to external data sources and specialized tools.

Prerequisites

uv is essential for running MCP servers in this lab. Ensure you have uv installed in your environment:

Installation Status Check:

```
uv --version
```



If not installed, follow these steps:

- **Windows:**

```
powershell -c "irm https://astral.sh/uv/install.ps1 | iex"
```



- **macOS/Linux:**

```
curl -Lsf https://astral.sh/uv/install.sh | sh
```



- **Alternative with pip:**

```
pip install uv
```



Note: If `uv` command not found, close the current terminal window and open a new one.

Add MCP Server to Kiro

Click the Kiro Ghost icon in the activity bar (left sidebar). In **MCP SERVERS** section, click the edit button or any tool name to insert a placeholder prompt in the chat. In this workshop, let's add Strands Agents MCP Server to provide curated documentation access to the complete Strands Agents documentation:


```
{  
  "mcpServers": {
```



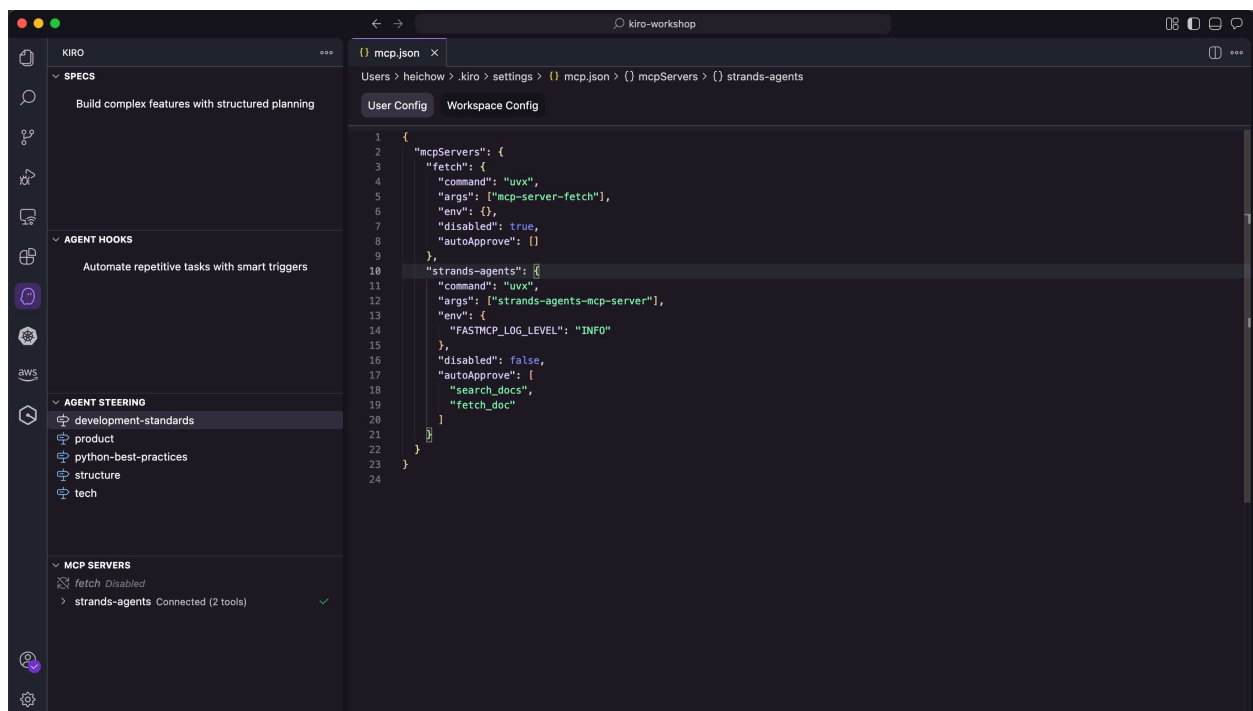
```

    "fetch": {
      "command": "uvx",
      "args": ["mcp-server-fetch"],
      "env": {},
      "disabled": true,
      "autoApprove": []
    },
    "strands-agents": {
      "command": "uvx",
      "args": ["strands-agents-mcp-server"],
      "env": {
        "FASTMCP_LOG_LEVEL": "INFO"
      },
      "disabled": false,
      "autoApprove": [
        "search_docs",
        "fetch_doc"
      ]
    }
  }
}

```

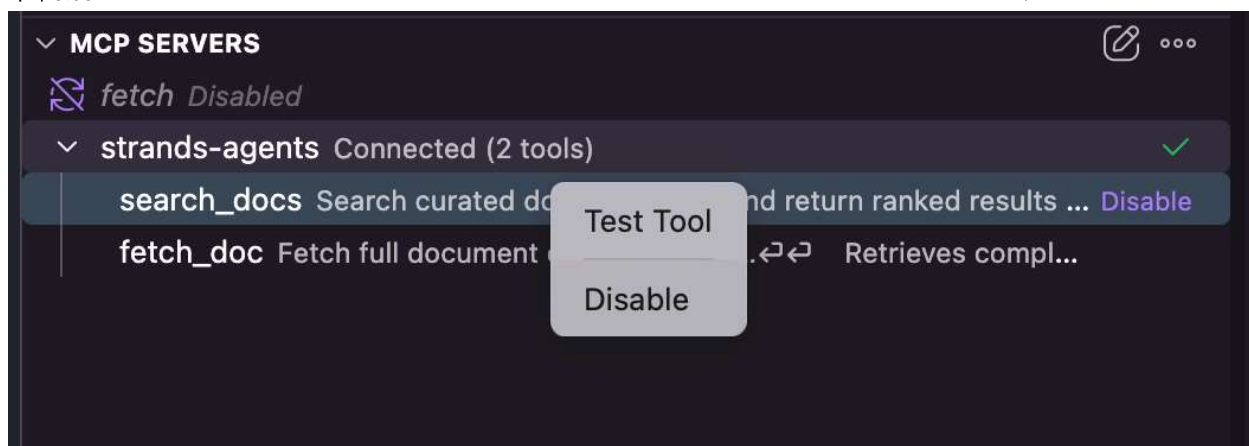
Save the `mcp.json` file. You should see the MCP Servers are Connected with  in the **MCP SERVERS** section.

Note: If Kiro fail to connect Strands Agents MCP Server, you may ask Kiro to troubleshoot by sending the prompt: Fail to connect strands-agents MCP server.

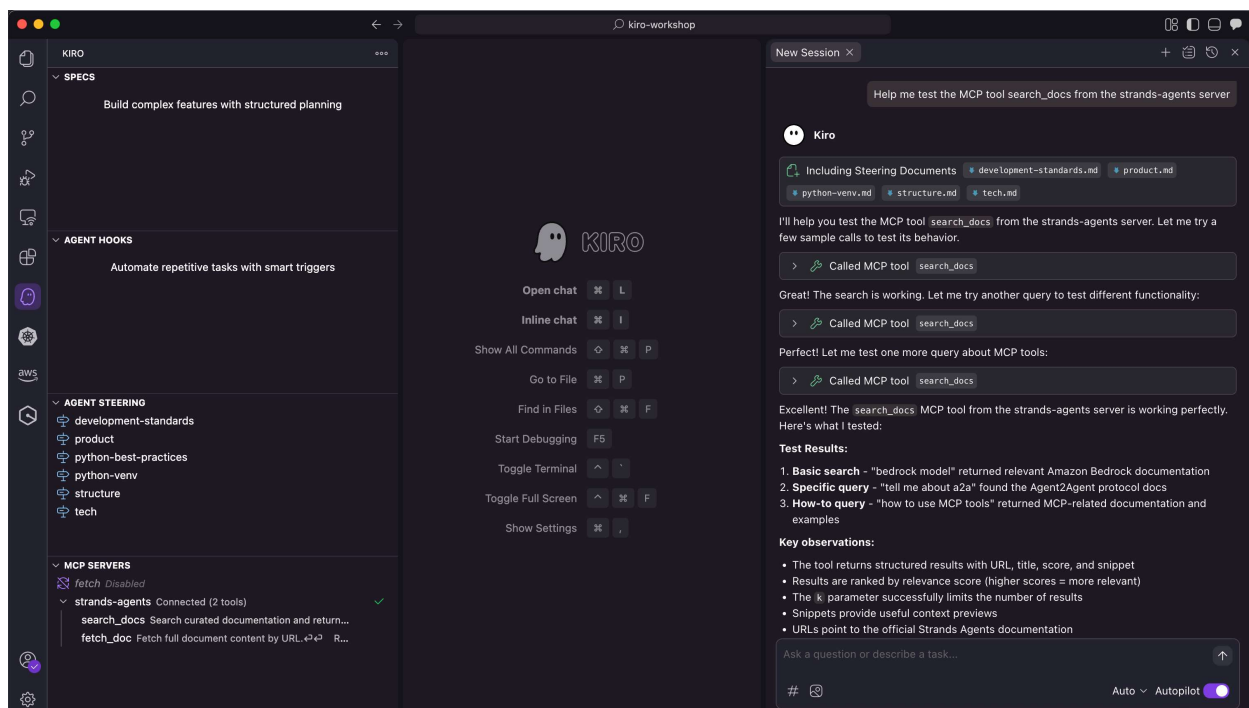


Test MCP Server

In the **MCP SERVERS** section, expand **strands-agents** and you will see there are two tools - `search_docs` and `fetch_doc`. Right click `search_docs` tool, then choose **Test Tool**.



The prompt `Help me test the MCP tool search_docs from the strands-agents server` is generated in Kiro's chat panel. Run the prompt to test the tool. Expand each **Called MCP tool** cell to see the tool use detail.



You can also test Strands Agents MCP Server by asking Strands Agents related question in Kiro's chat panel.

For example:


How to create a simple agent with Strands Agents?




Kiro has the knowledge of Strands Agents to answer your questions from searching and fetching Strands Agents documentations via MCP Server.

New Session ×

Let me search for API reference or basic agent concepts:

>  Called MCP tool `search_docs`

Let me get the full content of the multi-agent example to see actual code:

>  Called MCP tool `fetch_doc`

Based on the Strands Agents documentation, here's how to create a simple agent:

Creating a Simple Agent with Strands Agents

1. Installation

First, install Strands Agents:

```
# Standard installation
pipx install strands-agents-builder

# Or manual installation
pip install strands-agents
```

2. Basic Agent Structure

Here's the fundamental pattern for creating a simple agent:

```
from strands import Agent

# Create a basic agent
agent = Agent(
    system_prompt="You are a helpful assistant.",
    tools=[], # Add tools as needed
)

# Use the agent
response = agent("Hello, how can you help me?")
print(response)
```

MCP Examples

You may find other MCP examples for reference here:

- [AWS MCP Servers](#) 
- [Strands Agents MCP Server](#) 
- [GitHub MCP Server](#) 


© 2008 – 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

[Privacy policy](#)

[Terms of use](#)

[Cookie preferences](#)

Important Notes

Kiro currently supports local stdio MCP servers only, however you can add remote MCP servers by leveraging the [mcp-remote](#)  npm package to make requests to the remote MCP endpoint.

[Previous](#)[Next](#)