Back to Series

Claude Code
Features Guide

Part 6 of the Claude Code Series

Claude Code MCP Integration: Connect Your AI to Every Tool You Use

Learn how Claude Code uses the Model Context Protocol (MCP) to connect to databases, project management tools, analytics platforms, and hundreds of external services. A complete guide for teams and agencies.

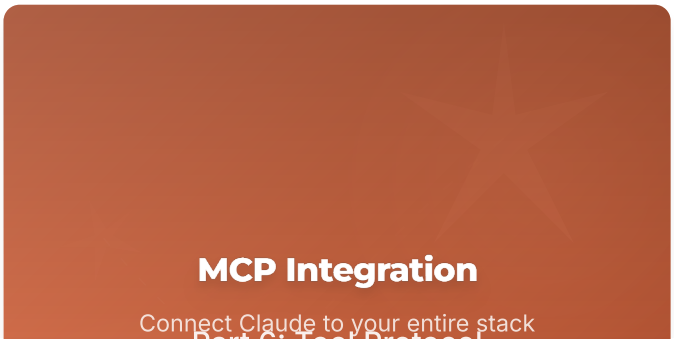
9 February 2026 12 min read

Claude Code is an exceptional coding assistant on its own. But coding does not happen in isolation. You need data from your database. You need context from your issue tracker. You need to check error logs, review designs, and push to deployment pipelines.

Without integrations, you end up copying and pasting between tools -- feeding Claude information it should be able to access directly. The Model Context Protocol (MCP) eliminates that friction entirely.

MCP is an open standard that lets Claude Code connect to external tools and data sources through a unified protocol. With MCP servers configured, Claude can query your database, read your GitHub issues, check your error monitoring, and interact with your project management platform -- all from within a single conversation.

This is what transforms Claude Code from a tool into a platform.



What Is the Model Context Protocol?

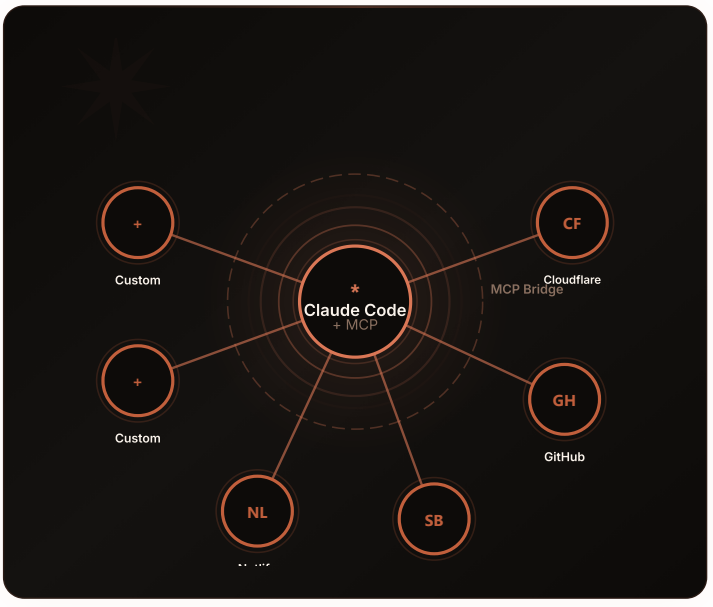
MCP is an open source standard developed by Anthropic for connecting AI assistants to external tools and data sources. Think of it as a universal adapter. Instead of building custom integrations for every tool, MCP provides a consistent interface that any AI assistant can use to communicate with any compatible server.

The protocol defines three core capabilities:

- **Tools**: Actions Claude can perform (query a database, create an issue, send a message)
- **Resources**: Data Claude can reference (files, database schemas, API documentation)
- **Prompts**: Pre-built commands exposed by MCP servers (list PRs, create tickets, run reports)

An MCP server is a lightweight program that exposes these capabilities for a specific tool or service. There are MCP servers for GitHub, Slack, PostgreSQL, Notion, Jira, Sentry, Google Drive, and hundreds of others. When you connect an MCP server to Claude Code, those capabilities become available in your conversation.

The MCP Ecosystem



How MCP Works in Claude Code

Adding a Server

Connecting a new MCP server takes a single command. There are three transport types depending on whether the server runs locally or remotely:

Remote HTTP servers (recommended for cloud services):

```
claude mcp add --transport http notion https://mcp.notion.com/mcp
```

Remote SSE servers (older protocol, still widely supported):

```
claude mcp add --transport sse asana https://mcp.asana.com/sse
```

Local stdio servers (for tools that run on your machine):

```
claude mcp add --transport stdio db -- npx -y @bytebase/dbhub \
  --dsn "postgresql://readonly:pass@prod.db.com:5432/analytics"
```

Once added, Claude has immediate access to the server's tools and resources. Ask Claude to query your database, and it uses the database MCP server. Ask it to create a GitHub issue, and it uses the GitHub MCP server.

Authentication

Many cloud-based MCP servers require authentication. Claude Code supports OAuth 2.0 for secure connections. After adding a server, run `/mcp` within Claude Code and follow the browser-based login flow. Authentication tokens are stored securely and refreshed automatically.

For servers that require pre-configured OAuth credentials (those that do not support dynamic client registration), you can provide your client ID and secret when adding the server:

```
claude mcp add --transport http \
  --client-id your-client-id --client-secret --callback-port 8080 \
  my-server https://mcp.example.com/mcp
```

Managing Servers

```
# List all configured servers
claude mcp list

# Get details for a specific server
claude mcp get github

# Remove a server
claude mcp remove github
```

```
# Check server status within Claude Code
/mcp
```

Model Context Protocol

MCP is a standardised protocol that lets AI tools talk to external services — databases, APIs, cloud platforms — through a secure, structured interface.

Why MCP Matters: Real-World Examples

The power of MCP becomes clear when you see it in action.

Implement a Feature from Start to Finish

With GitHub and Sentry MCP servers connected, you can say:

> *""Add the feature described in JIRA issue ENG-4521 and create a PR on GitHub.""*

Claude reads the issue, understands the requirements, implements the code, and opens the pull request -- without you copying a single piece of context between tools.

Debug Production Issues

> *""What are the most common errors in Sentry over the last 24 hours? Show me the stack traces and suggest fixes.""*

Claude queries your error monitoring directly, analyses the patterns, cross-references with your codebase, and proposes solutions.

Query Business Data

With a PostgreSQL MCP server:

> *""Find our top 10 customers by revenue this quarter and show me their usage patterns.""*

Claude writes and executes the query, then analyses the results -- all within the same conversation where you are making product decisions.

Automate Cross-Tool Workflows

> *""Check Slack for the latest design feedback, update the Figma components accordingly, and create a GitHub issue for anything that needs engineering work.""*

Each tool in the chain is accessed through its MCP server. Claude orchestrates the workflow while you focus on decisions.

4+

Live MCP Connections

Cloudflare (DNS, Workers, KV), GitHub (repos, PRs), Supabase (agent database), and Netlify (serverless functions) — all accessible from within Claude Code.

MCP Tool Search: Scaling Without Context Bloat

Here is a practical problem. Every MCP server exposes tool definitions that Claude needs to understand. If you have ten servers with twenty tools each, that is two hundred tool definitions competing for space in Claude's context window.

MCP Tool Search solves this elegantly. When your MCP tool descriptions would consume more than 10% of the context window, Claude Code automatically enables on-demand tool loading. Instead of preloading every tool definition, Claude uses a search mechanism to discover relevant tools only when it needs them.

From your perspective, nothing changes -- Claude still has access to every tool. But behind the scenes, only the tools Claude actually uses are loaded into context. In real-world usage, this has shown a 46.9% reduction in total agent tokens, dropping from 51,000 tokens to 8,500 for typical workflows.

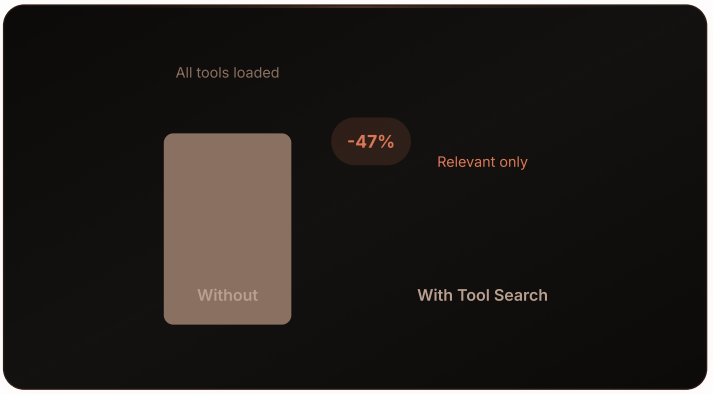
You can configure the threshold or disable tool search entirely:

```
# Custom 5% threshold
ENABLE_TOOL_SEARCH=auto:5 claude

# Always enabled
ENABLE_TOOL_SEARCH=true claude

# Disabled
ENABLE_TOOL_SEARCH=false claude
```

Tool Search: Token Efficiency



MCP Server Scopes: Personal, Project, and Organisation

Where you configure an MCP server determines who can use it:

Local Scope (Default)

Stored in your personal configuration. Private to you, available only in the current project. Use this for personal development servers or configurations with sensitive credentials.

```
claude mcp add --transport http stripe https://mcp.stripe.com
```

Project Scope

Stored in a `.mcp.json` file at your project root, designed to be committed to version control. Every team member who clones the repository gets the same MCP configuration.

```
claude mcp add --transport http paypal --scope project https://mcp
```

This is significant for teams. Your entire toolchain configuration travels with the project. New team members -- or new AI agents -- have immediate access to every integration the project needs.

User Scope

Stored in your personal configuration but available across all your projects. Use this for utilities you rely on everywhere, like a personal database or notes system.

```
claude mcp add --transport http hubspot --scope user https://mcp.hu
```

The Agency Angle: Connecting Claude to Your Client's Toolchain

For agencies managing multiple clients, MCP is a force multiplier. Each client project can have its own `.mcp.json` file with the exact integrations that project requires.

SEO and Analytics

Connect Claude to Google Search Console, Google Analytics, and rank tracking tools. Instead of exporting CSVs and pasting data into prompts, Claude queries the live data directly. Ask it to analyse keyword performance, identify traffic drops, or compare month-over-month metrics -- and it pulls the numbers in real time.

Content Management

CMS integrations let Claude read existing content, check publication schedules, and understand the content landscape before writing new material. For WordPress, Contentful, Sanity, or any CMS with an MCP server, Claude becomes content-aware.

Project Management

Jira, Linear, Asana, and other project management tools expose their data through MCP. Claude can read sprint backlogs, create tickets from code analysis, update task statuses, and link implementation work to the issues that requested it.

Error Monitoring and Observability

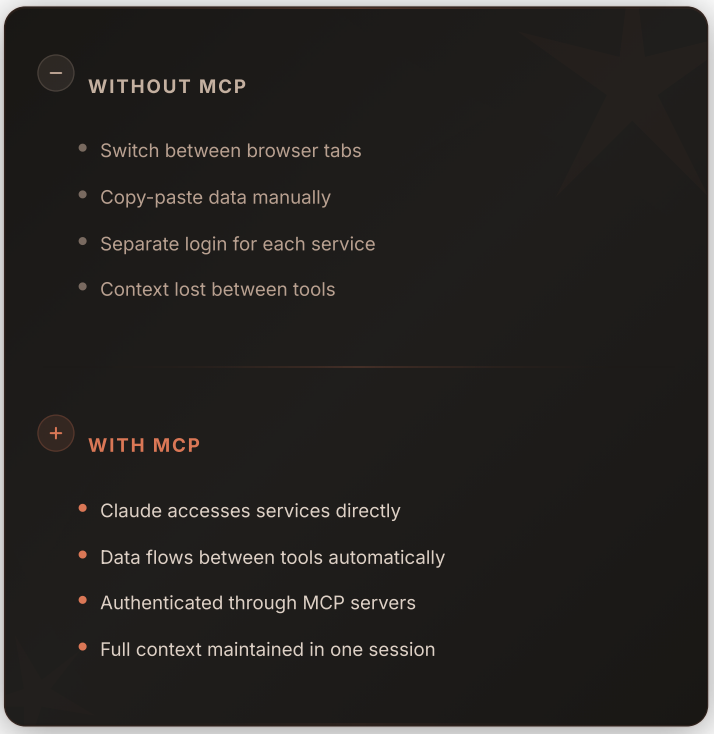
Sentry, Datadog, and similar platforms expose error data through MCP. When debugging, Claude can check production errors, correlate them with recent deployments, and propose fixes based on actual stack traces rather than hypothetical scenarios.

The Compound Effect

The real power is combining multiple integrations in a single workflow. An agent running an SEO audit might:

1. Pull live search performance data via Google Search Console MCP
2. Check current site structure via a CMS MCP
3. Cross-reference with analytics data for user behaviour patterns
4. Create prioritised tasks in the project management MCP
5. Report findings in the team's Slack channel

Each step uses a different MCP server, but to Claude it is one continuous workflow. No context switching, no copy-pasting between tools, no lost information.



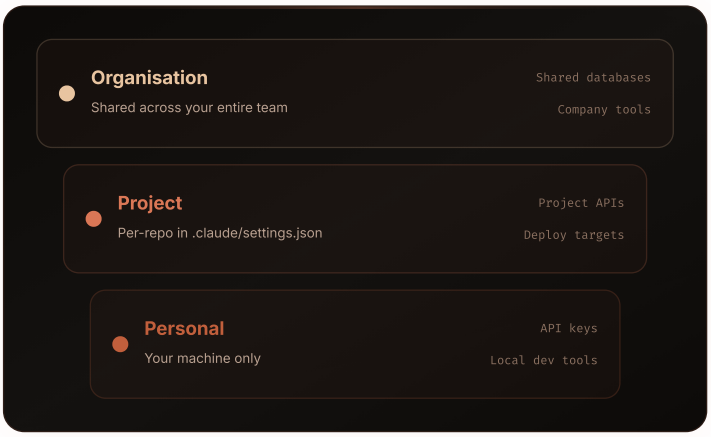
Enterprise and Managed MCP

For organisations that need centralised control, Claude Code supports managed MCP configuration. Administrators can deploy a `managed-mcp.json` file that defines exactly which MCP servers are available -- and prevents users from adding unauthorised servers.

You can also use allowlists and denylists to set policy-based restrictions: allow users to add their own servers, but only from approved domains or using approved commands.

This level of control matters for agencies handling sensitive client data. You can ensure that client credentials flow only through approved integrations, that production databases are accessed read-only, and that every MCP connection meets your security requirements.

Three MCP Scopes



Key Takeaways

1. **MCP is an open standard** for connecting AI assistants to external tools and data sources
2. **One command to connect**: `claude mcp add` with HTTP, SSE, or stdio transport
3. **Tool Search scales intelligently**, loading tools on demand to preserve context
4. **Three scopes** (local, project, user) let you control who sees which integrations
5. **OAuth 2.0 authentication** keeps connections secure with automatic token refresh
6. **Project-scoped** `.mcp.json` files make integrations travel with your codebase
7. **Hundreds of pre-built servers** cover databases, APIs, project management, and more
8. **Enterprise controls** let organisations manage which servers are permitted

MCP Server Configuration

terminal

```
// .claude/settings.json
{
"mcpServers": {
"supabase": {
"command": "npx",
"args": ["-y", "@supabase/mcp-server"]
},
"github": {
"command": "npx",
"args": ["-y", "@github/mcp-server"]
}
}
}
```

MCP servers are configured declaratively. Each server provides tools that Claude can call — no manual API integration required.

Ready to Connect Your AI to Your Toolchain?

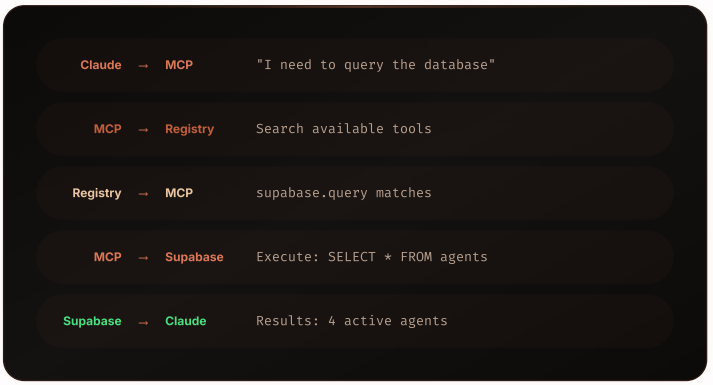
Jordan James Media helps Australian businesses integrate AI into their existing workflows. From MCP configuration to full autonomous agent systems, we make sure your AI works with your tools -- not around them.

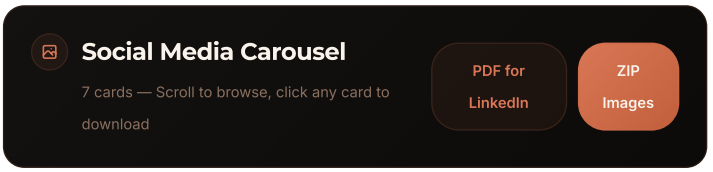
Talk to Us About AI Integration

Related Reading:

- Claude Code's Biggest Update: 7 Features That Changed How We Build
- Claude Code Skills: How to Codify Your Agency's Entire Workflow
- Claude Code Background Tasks: Your AI Works While You Sleep

How Tool Discovery Works





Share This Article
Spread the knowledge

Twitter LinkedIn Copy

Build With Claude Code

We use Claude Code to ship faster, smarter, and at scale. Let us build your next project.

Get Your Free AI Strategy → App Development

Claude Code
8 parts

View All

Quick Links

→ Series Home
→ App Development
→ Get In Touch

Claude Code Stats

Model	Opus 4.6
Features	7
Series Parts	8

Agency Tested
Real projects, real results

Ready to outperform your competition?

Get your free AI strategy consultation today.

Start Your Free Consultation →

Jordan James Media
GROWTH STUDIO

Australia's AI-first digital marketing agency. We build intelligent systems that drive exponential growth.

info@jordanjamesmedia.com

0475 897 737

Sydney, Australia

SERVICES

- SEO Services
- Google Ads
- Website Design
- Local SEO
- Content Marketing
- App Development

COMPANY

- About Us
- Blog
- Case Studies
- Pricing
- Tools
- Contact

LOCATIONS

- Sydney
- Melbourne
- Brisbane
- Perth
- Adelaide

All Locations

All Locations →

Built with Claude 4.6 Opus