MCP · NitroStack · NitroStudio · NitroCloud



BUILDATHON GUIDEBOOK

Reimagine the products you know

Build the experiences of tomorrow

Model Context Protocol (MCP)

Introduced by Anthropic in 2024, MCP is an open standard that has been defining the future of AI ever since - bridging the gap between isolated AI models and real-world software.

The Future Is Agentic

- **MCP is a standard for connecting AI agents to external applications.** It gives AI a common way to interact with the software application people already use, moving AI from simply answering questions to actually getting things done.
- **Imagine telling your Agent,** *I have two hours free tonight - plan something for me***.** Through MCP, the AI could connect to your calendar, find a movie that fits, book the tickets, and add it to your schedule - all as one connected experience.

Before You Build

You don't need to become an MCP expert before the buildathon. Build a mental model, play around with MCP once, and learn the deeper details as you build.

- modelcontextprotocol.io/docs The ultimate source of truth for the protocol. You do not need to read the entire documentation but it is a great way to build your mental model and explore the many aspects of MCP.
- Connect Me This is a quick task for you - connect your first MCP server to ChatGPT and play around with it.
- MCP in Action Try the MCP Inspector to connect to an MCP server and inspect its capabilities locally

Ideation

The Four Steps to a Real Idea

- Pick a well-known product. Choose software that people already use and can easily recognize.
- Find one workflow to reimagine. Focus on a specific experience within that product.
- Identify what is missing. Find what is slow, manual, frustrating, or impossible today.
- Reimagine it with MCP and NitroStack. Show how MCP can make that workflow meaningfully better, smarter, or capable of something the original product cannot do today

Worked Example

- **Starbucks:** You want to get your usual coffee, but every time you have to open the app, find your order, place it, choose a store, and wait for it to be ready.
- **Reimagined with MCP:** An AI could remember your preferences, understand where you're going, connect with Starbucks, and place your order at the right time - turning a routine task into a simple conversation.

Open Innovation

There are no predefined tracks or themes. Choose any well-known product, application, or developer tool and reimagine a workflow in a way that creates something genuinely new.

What Not to do

- Don't pick an unfamiliar product. Choose something well-known and impactful.
- Don't recreate what already exists. Your idea should meaningfully improve the original experience.
- Don't clone the original product. Use mock data where needed and focus on your new experience.
- Don't make MCP an afterthought. If your idea works the same without MCP, rethink it.
- Don't build below your potential. **Think bigger, challenge yourself, and build something worth showing**.

NitroStack

Now that you have the context on MCP and what you need to build, meet NitroStack, the end-to-end platform for building, testing, and deploying MCP applications.

NitroStack takes care of much of the infrastructure around MCP, so you can focus on building the experience and capabilities that make your idea useful.

Your MCP Build Toolkit

- **NitroStack SDK** - A TypeScript framework for building production-ready MCP servers with a simple, structured developer experience.
- **NitroStudio** - Your development and testing environment. Connect your local MCP server, test its tools, inspect results, and interact with it through an AI-powered chat interface.
- **NitroCloud** - Deploy your MCP application to the cloud and turn your local project into a live, accessible MCP server.
- **NitroChat** - Connect your MCP integrations to a public AI experience, allowing people to interact with the capabilities you build.

Where to Learn More

- **Watch:** Check out the NitroStack YouTube playlist for an extensive walkthrough covering everything you need to get started with NitroStack and begin building. Watch NitroStack 101
- **Read:** A visual, written handbook covering the key concepts and steps for building with NitroStack. If you want to get a head start with a reference guide beside you, NitroStack Handbook
- **Explore:** Dive deeper into the documentation for a broader overview of NitroStack and the capabilities available to you as you build. Explore the NitroStack Docs

The Pieces You'll Work With

- **Tools** - Actions your MCP server exposes for an AI agent to discover and execute
- **Tasks** - Asynchronous operations for work that may take time to complete, allowing the agent to start a task and check its progress or result later
- **Resources** - Information your MCP server makes available for an AI agent to read and use as context.
- **Prompts** - Reusable instructions that help guide how an AI agent interacts with your MCP server.
- **Widgets** - Interactive UI components that let your MCP tools return rich, visual experiences instead of plain text.
- **Guards** - Security controls that protect your MCP server and its capabilities by managing who can access or execute them.

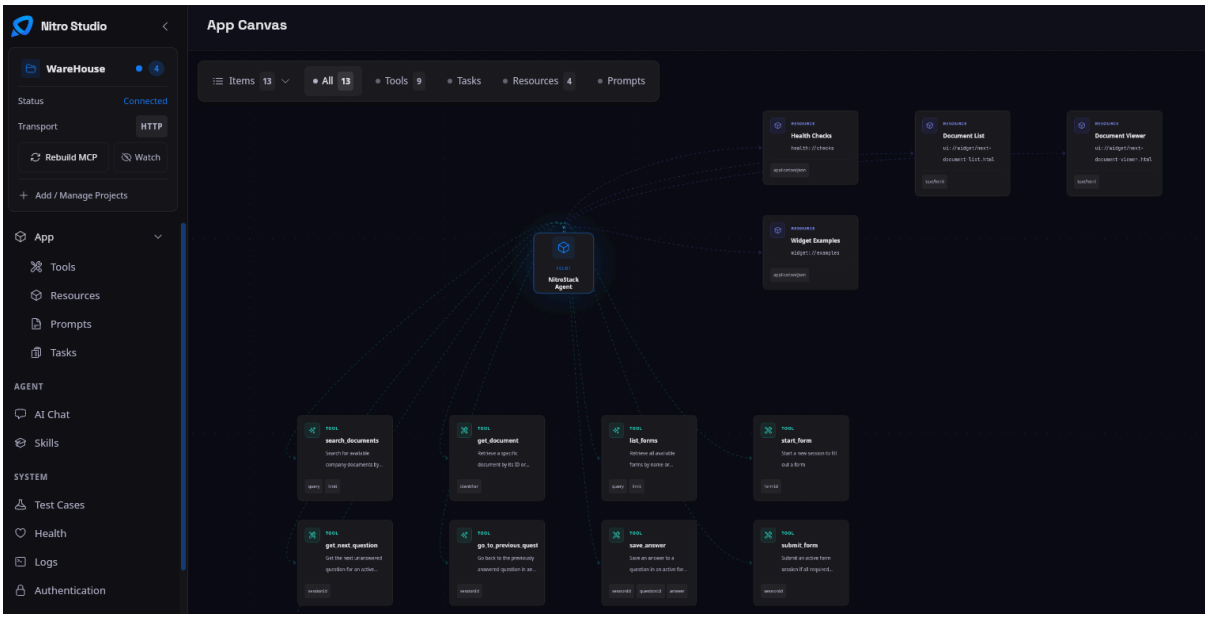
NitroStack SDK

Get started by installing the NitroStack SDK from NitroStack SDK and using the suitable template to set up your MCP project. Build and customize your server around the workflow you want to reimagine.



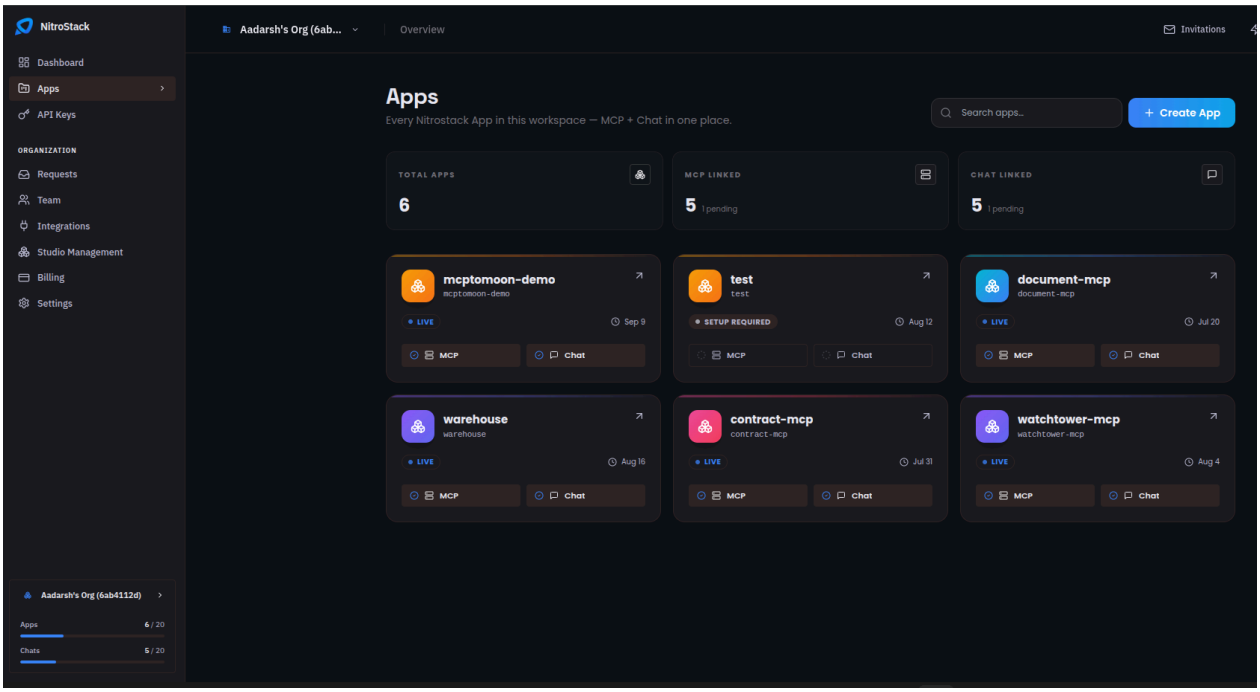
NitroStudio

Install NitroStudio, available for Windows, macOS, and Linux, to connect your MCP server and test everything you build. Use it to inspect your tools, resources, prompts, tasks, and widgets and interact with your server through AI chat.



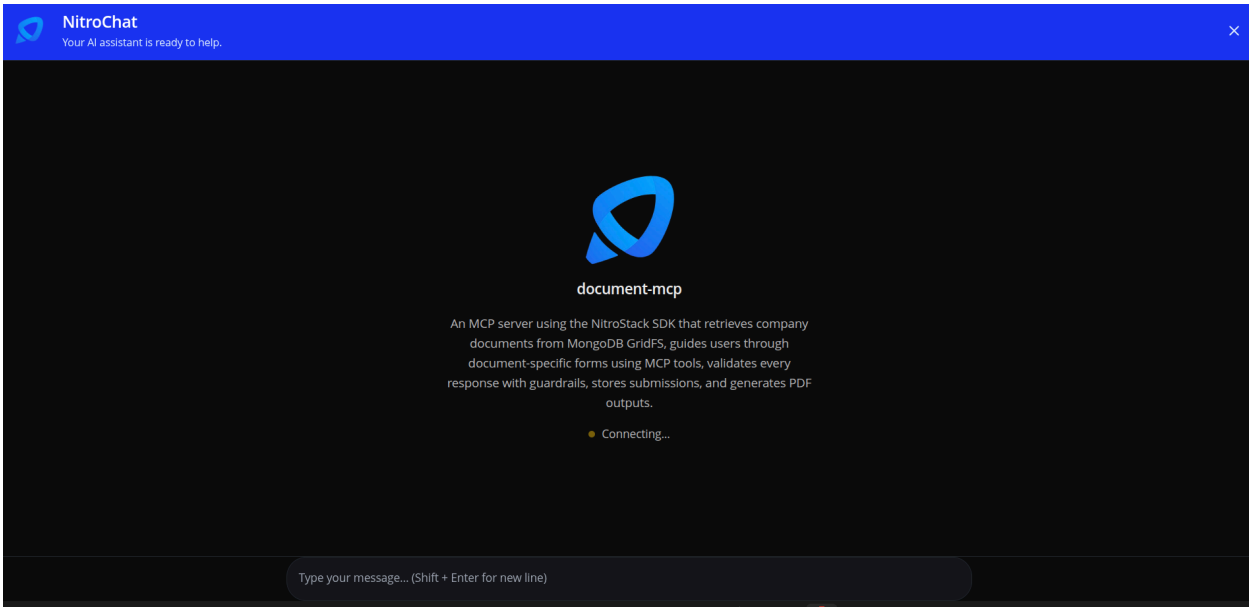
NitroCloud

Create your NitroCloud account at NitroCloud to start deploying your MCP application. Deploy your project to the cloud and turn your local MCP server into a live, accessible application.



NitroChat

Use NitroChat to create a public AI experience around your deployed MCP application. It lets others interact with the capabilities and agent experience you have built.



Submission

Your project is not finished until it is submitted. Build, test, share, and make sure everything required is ready before you submit.

Submission Portal

- All projects must be submitted through the dedicated submission portal. The portal is the official submission channel for the buildathon.
- Your submission will only be considered complete once all required project specification and social links have been provided. More details about the portal and submission process will be shared during the hackathon.

What You Need to Submit

- **Project Name** - Give your project a clear and memorable name.
- **Original Product** - Name the product, platform, or developer tool you reimagined.
- **Project Description** - Briefly explain the workflow you reimagined and what you changed.
- **GitHub Repository** - Provide a link to your project repository.
- **Demo Video** - Include a link to your project demo video.
- **Social Posts** - Include links to your project posts on YouTube, X, and LinkedIn.

Demo Video

Your demo video plays an important role in judging. It is your chance to showcase your product and create an impact in the minds of the judges. Be creative with how you present it, put effort into the quality and editing**.** Keep it **under 3 minutes**.

Social Posts

- Sharing your project on **LinkedIn, X, and YouTube is mandatory**. Create a post for your project on each platform, tag the official **NitroStack handles** in every post, and include the links to all three posts in your submission.
- This is an opportunity to showcase your work publicly, build your presence, and reach a wider audience beyond the buildathon
- NitroStack on LinkedIn
- NitroStack on Youtube
- NitroStack on X

Before You Hit Submit

- Test your live deployment and make sure your project works as expected.
- Watch your demo video and confirm that it clearly shows the project in action.
- Check your GitHub repository and make sure it is accessible and includes a brief README.
- Verify all three social posts are live, include the required NitroStack tag, and that their proper links are included in your submission.

Support

Discord Is the Primary Channel

- All official support is communicated through Discord.
- Join as soon as you register, not after the event starts.
- Discord Invite Link

Ticket-Based Support

- High-priority, blocking issues go through a ticket-based system - opening one means our team reaches out to you directly.
- Use a ticket for, NitroStack setup problems, NitroStudio workflow issues, deployment or NitroCloud problems, and submission portal or account access issues.
- Please avoid using a ticket for casual questions or idea validation. The General or Help channels are the best places for these discussions.

Discord Channels

- **Announcements** - official updates, deadlines, and schedule changes.
- **General / Help** - casual discussion and questions that don't need a ticket.
- **NitroStack Help** - community discussion specific to NitroStack.

Frequently Asked Questions

Q. Do I need MCP experience before joining?

A. No. The NitroStack 101 playlist covers what you need to get started. Basic TypeScript and command-line familiarity are enough.

Q. Is there a specific theme I should follow?

A. No. The sky is the limit. Choose any idea you want, but make sure it reimagines a well-recognized application, software, or developer tool.

Q. How many members can be on a team?

A. This is a strictly solo hackathon. Learn, build, grow, and make the most of the experience on your own.

Q. Is posting on social media mandatory?

A. Yes. You must post your project on YouTube, X, and LinkedIn and tag the official NitroStack handles. Building in public helps you showcase your work, grow your presence, and get your project in front of a wider audience.

Q. What happens if I hit NitroChat's credit limit while demoing?

A. Fall back to your existing ChatGPT or Claude setup to demonstrate your MCP server working within an agentic workflow - the same setup is worth exploring early anyway, not just as a fallback.

Glossary

MCP - Model Context Protocol, an open standard that allows AI applications to connect with and interact with external systems.

Agent - An AI system that can understand a goal, make decisions, and take actions to complete it.

MCP Server - The application that makes capabilities available to an AI agent through MCP.

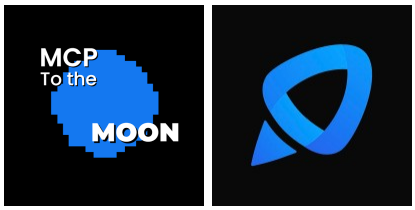
MCP Client - The application that connects to an MCP server and uses the capabilities it provides.

Agentic Workflow - A workflow where an AI agent can decide and execute multiple actions to accomplish a goal.

Before You Go

A last-pass checklist pulling together the most important points from every section above. If every box here is genuinely true, you're ready to start.

- Your idea is one specific workflow inside one recognized product, not the whole product.
- You can explain why MCP specifically makes your idea possible, in one or two sentences.
- NitroStack CLI is installed and your project is scaffolded.
- NitroStudio is installed, you're signed in, it's connected to your project, and every tool has been tested individually.
- Your NitroCloud account is set up well before you need to deploy.
- You know what a ticket is for, and what it isn't for, in Discord.
- You have a rough plan for your five days, even if it shifts as you go



You've got the whole picture now.

Go pick something worth fixing, and build it.

See you on Discord.