

Tracing Claude Code to LangSmith

4 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=W_XP06VT6AU](https://www.youtube.com/watch?v=W_XP06VT6AU)

https://www.youtube.com/watch?v=w_xp06VT6aU

SUMMARY

To set up tracing from Claude Code to Langmith, users need to create a Langmith account, generate an API key, and configure several environment variables for telemetry. This setup allows Claude Code to emit OpenTelemetry events, which Langmith collects for monitoring and observability.

- *Create a Langmith account and generate a personal access token for API access.*
- *Enable telemetry in Claude Code by setting the environment variable `CLOUD_CODE_ENABLE_TELEMETRY=1`.*
- *Specify the log exporter format as OpenTelemetry Protocol (OTLP) and configure HTTP transport with JSON encoding.*
- *Set the appropriate Langmith endpoint based on whether using Langmith Cloud or a self-hosted instance.*
- *Include the API key in headers for authentication and specify a tracing project for organization.*
- *Enable logging of user prompts and inputs by setting the corresponding variable to true.*
- *Monitor and analyze traces in Langmith, including token usage, latency, and operational details from Claude Code sessions.*
- *Utilize pre-built dashboards in Langmith to track API usage patterns and costs over time.*

Hey there. Today you're going to learn how you can set up tracing from Claude Code to Langsmith by just setting a few environment variables. Let's dive right in. First, you're going to need to create a Langsmith account and generate an API key. Once you have an account, navigate to your settings page and create a personal access token. This is the only key that we need to connect Claude Code to Langsmith. Let's copy this and we'll use it in a second. Claude Code is one of the most impressive and useful AI coding tools to date. I use it every day in my work including on this open deep research agent which I have open here. What's nice is that cloud code can emit open telemetry standard events for monitoring and observability.

Langsmith can then collect and display these events to give you a full detailed log on everything that happens during your cloud code sessions. To set up tracing to Langsmith, all we need to do is set a few environment variables to configure the hotel export from cloud code. The first environment variable that we're going to set is cloud code enable telemetry equal to 1. This basically turns on telemetry for cloud code. Next, we'll specify hotel logs exporter equals OTLP. This specifies the output format to use open telemetry protocol. We'll also specify that the log should be sent with HTTP transport and JSON encoding. This is the format that Linksmith ingestion was built to accept. Now, this is the key piece. If you're using Langsmith Cloud like I just showed, this is the endpoint that you'll specify for logs from Cloud Code. If you're using a self-hosted instance of Langsmith, your URL will look a little bit different. You can refer to the docs in the description to go get your URL.

Now, we're going to take that API key that we generated earlier and set that in our headers. This is going to allow us to authenticate and connect to Langsmith. We'll also specify a tracing project. This will specify where our cloud code traces show up in Langsmith so we can find them easily. Finally, we want to enable logging of user prompts and inputs. So, we'll set this variable to true. That's it. All we needed to do was set these six environment variables.

Claude Code is now emitting events that are captured and displayed in Linksmith. Now, let's see it in action. Let's go ahead and start a Claude Code session. I'm going to ask Claude Code to broadly describe to me what Open Deep Research does and how it works. We can see that Claude Code is reading quite a few files. This will take a while, so I'm going to skip ahead to when the answer has been generated. We can see that this response is pretty helpful. Let's see what we traced in Langsmith.

We can see a new trace here in Langsmith named Claude Code. If I click into it, I can see each of the individual things that Claude Code did. I've logged the question that was asked initially. We can also see the model names, token usage, and latency from the model requests that Claude Code makes. Cloud code also sends up costs associated with each request. And we can also see all of the operations that Claude Code undertakes, like reading files.

Now, let's ask a follow-up. We're going to ask Claude Code to write a cloud.md file for this repo. We can see that in this case it comes up with a to-do list and asks us to execute a few commands. We'll play along with it here and go with it. Now let's check back in on our trace in Langsmith. In that same trace, we can see that we now have more runs which correspond to the different actions that Claude Code took. One thing to note here is that while we can see the user prompts that we input, we don't have access to the actual system prompts and messages that Claude Code sends to the Anthropic models. We also can't see the raw model outputs that come back. This information isn't exported by Anthropic's hotel logging. However, we do get token count and cost measurements. We can actually see that after the second question, the total token count and cost for this trace has gone up. This is because each Claude Code trace is tied to a session. In other words, everything that I do in this session of Claude Code is going to get logged to this trace. The waterfall view is particularly interesting. We can already see groups of runs based on timestamp for our first user prompt asking about a repo and then our second user prompt asking Claude Code to write a file.

Tracing Claude Code to Linksmith can also be really useful for organizations trying to monitor general usage. API usage for Claude Code can easily match or exceed that of a production application. Langsmith has these pre-built dashboards to help us see the total number of traces over time in any tracing project, as well as stay on top of any patterns in token usage or costs. To recap, you can set up tracing from Claude Code to Linksmith just by setting a few environment variables. I've linked a doc in the description with written instructions and those exact variables spelled out. Thanks for watching.