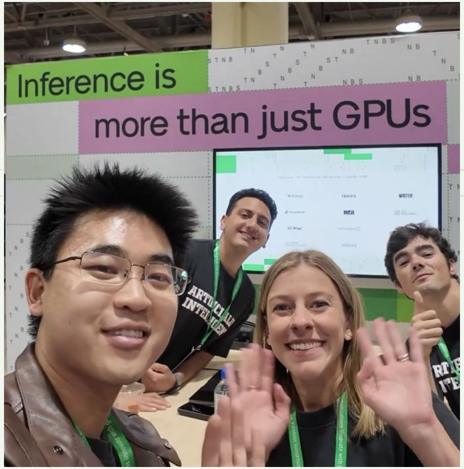
My AI Engineering Tech Stack

(How Open-Source Models ***Actually*** Run AI Coding at Scale)

Alex Ker
Engineer, Baseten

Hi, I’m Alex



And I work on AI infrastructure with these awesome + kind people at Baseten

> Previously built ML pipelines at Neurable (BCI) and RL at LaunchDarkly
> Founded an AI incubator p-ai.org and contributed to Stanford HAI as an editor
> I like calisthenics and reading in my free time

ABRIDGE
ambp
Bland®
clay
CURSOR
Decagon
descript
GAMMA
HeyGen
MERCOR
Mirage
Notion
OpenEvidence®
World Labs
WRITER

Some of our customers <3, building at the frontier

We get to partner with them to implement and create the patterns around agentic systems today

Agenda

1. Limitations of closed-source models
2. Rise of open-source coding models
3. How to use OSS in your dev workflows
4. Performant inference for coding models
5. Takeways for AI native builders

The problem with closed models

While closed source models bring a lot of advantages and out-of-the-box support and are an easy solution to get started, problems appear at scale.

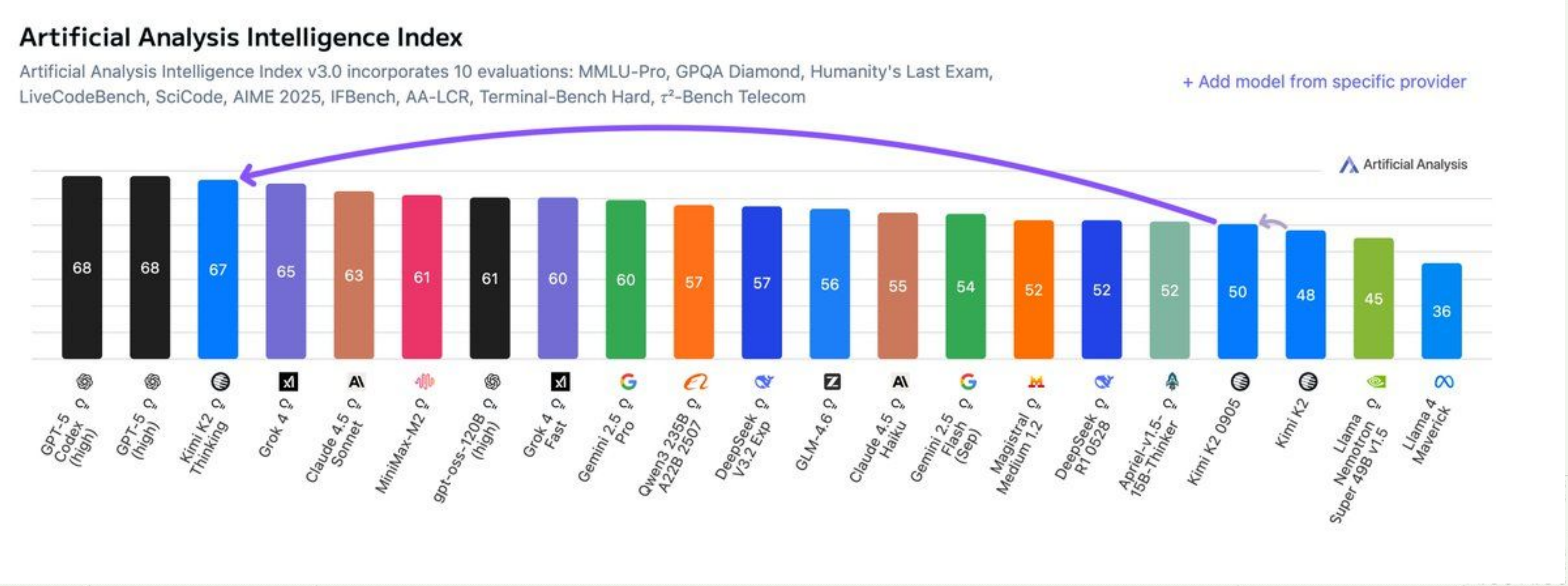
- **Rate limits** - TPS
- **Cost** - multiples more expensive
- **Reliability** - Unmet SLAs
- **Speed** - Throughput
- **Control** - Fine-tuning, transparency

Why use closed models?

Because they're smart!

But the quality gap is closing

As we recently saw the Kimi K2 Thinking drop, which is considered the smartest open-source model.
It's benchmark scores are competitive with models like GPT-5 and Claude Sonnet 4.5.



Artificial Analysis Intelligence Index

Artificial Analysis Intelligence Index v3.0 incorporates 10 evaluations: MMLU-Pro, GPQA Diamond, Humanity's Last Exam, LiveCodeBench, SciCode, AIME 2025, IFBench, AA-LCR, Terminal-Bench Hard, τ^2-Bench Telecom

+ Add model from specific provider

Advantages of Open Source

Latency (developer experience)

Reliability (uptime at peaks)

Cost (per token)

What does AI native product engineers care about?

- It turns out…the same things.
  - People care about these principles in their devtools AND product.
- Central to building production-ready apps with standout UI/UX within a competitive market.

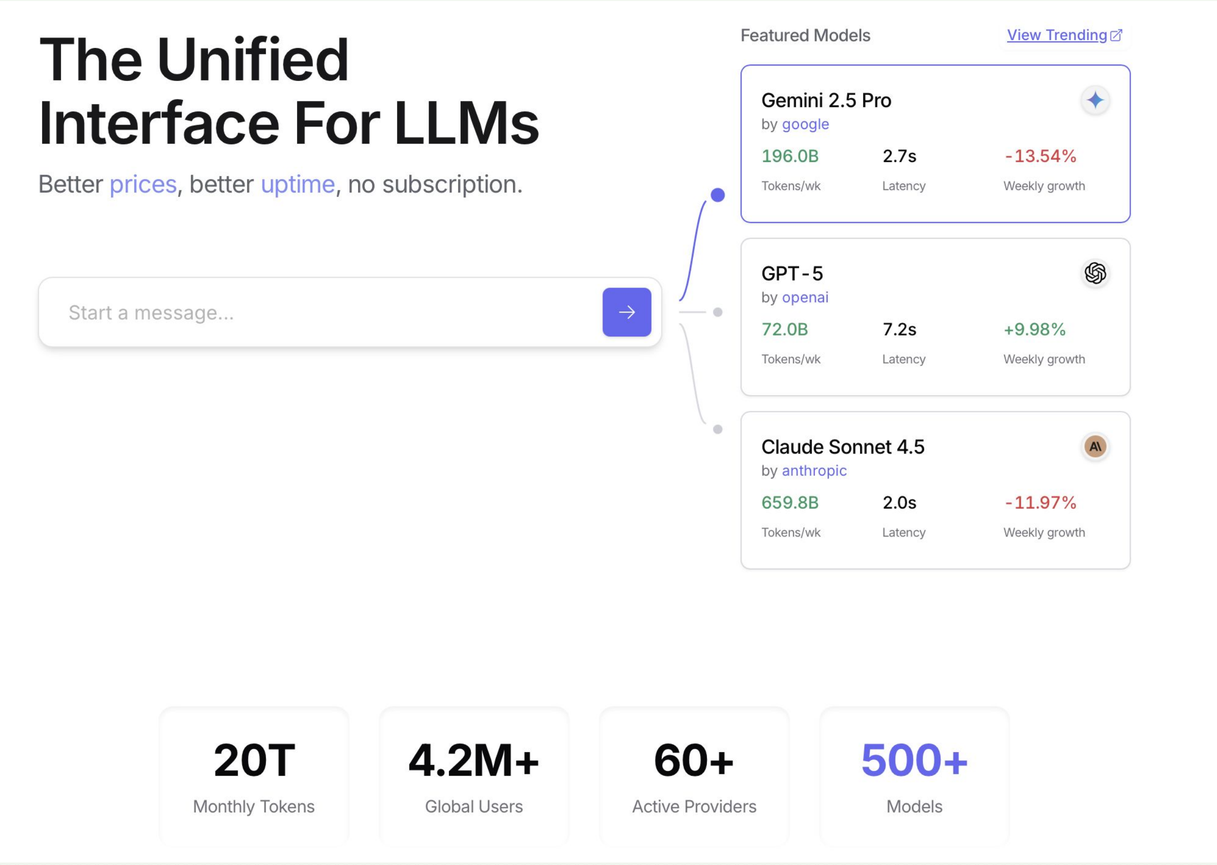
In 2025, the top 3 things we learned at Baseten:

1. **Reliability** is the new moat
2. **Speed** is the product
3. **Developers** want the wheel

The best open source models today

OpenRouter

- Unified access to 500+ models
- Automatic fallback for reliability
- Transparent metrics for providers



The Unified Interface For LLMs

Better prices, better uptime, no subscription.

Start a message...

Featured Models   View Trending

Gemini 2.5 Pro
by google

196.0B	2.7s	-13.54%
Tokens/wk	Latency	Weekly growth

GPT-5
by openai

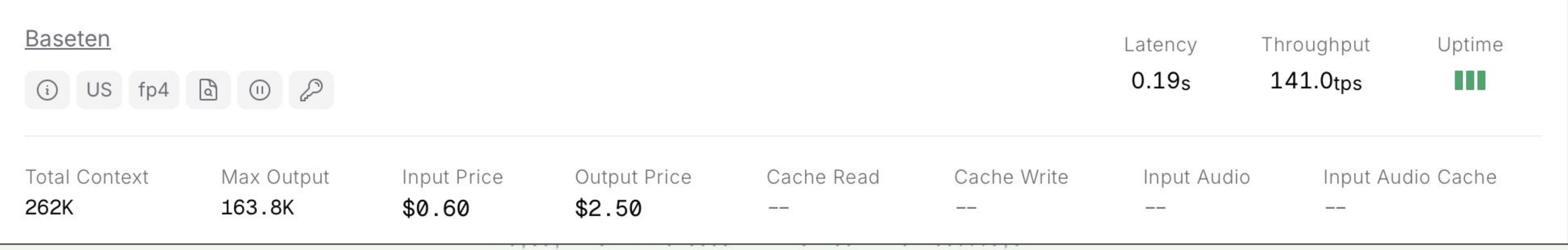
72.0B	7.2s	+9.98%
Tokens/wk	Latency	Weekly growth

Claude Sonnet 4.5
by anthropic

659.8B	2.0s	-11.97%
Tokens/wk	Latency	Weekly growth

20T	4.2M+	60+	500+
Monthly Tokens	Global Users	Active Providers	Models

GLM 4.6 on OpenRouter



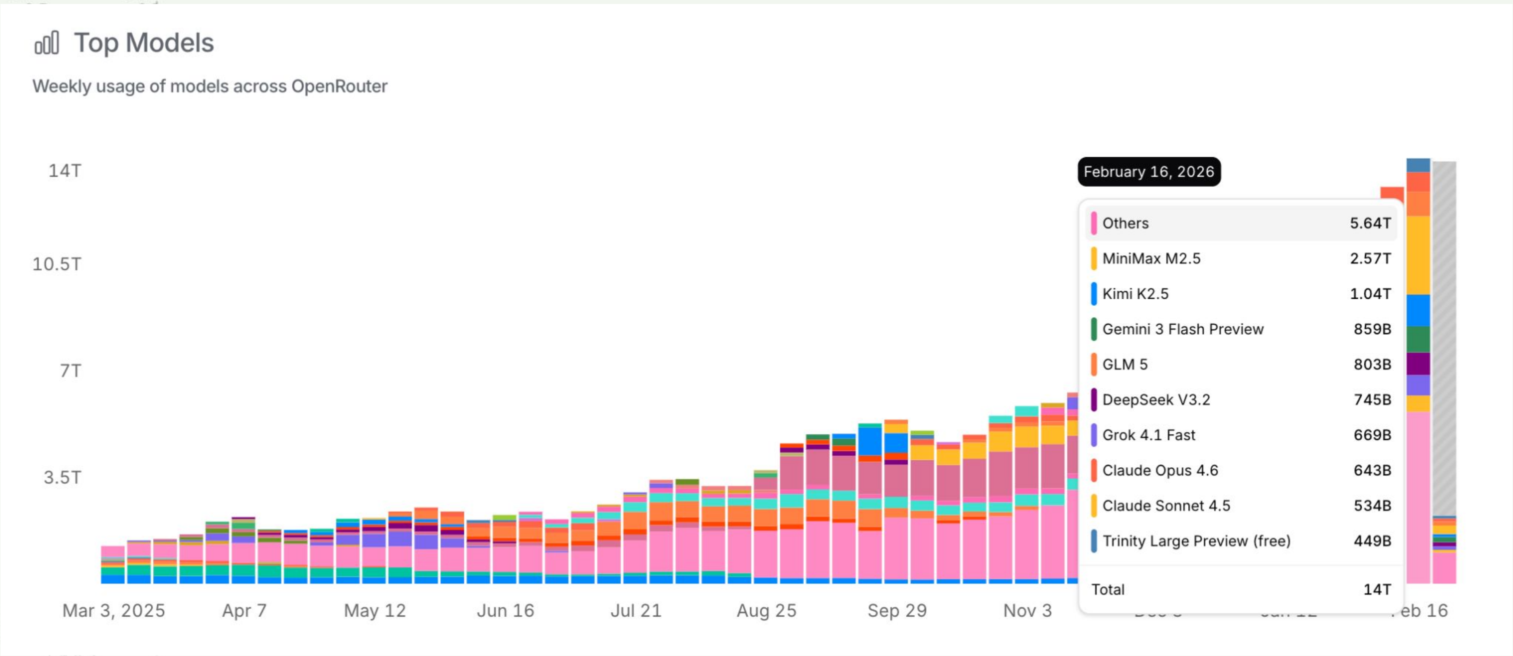
Baseten

US fp4

Latency	Throughput	Uptime
0.19s	141.0tps	🟩🟩🟩

Total Context	Max Output	Input Price	Output Price	Cache Read	Cache Write	Input Audio	Input Audio Cache
262K	163.8K	$0.60	$2.50	--	--	--	--

OpenRouter Stats 1)



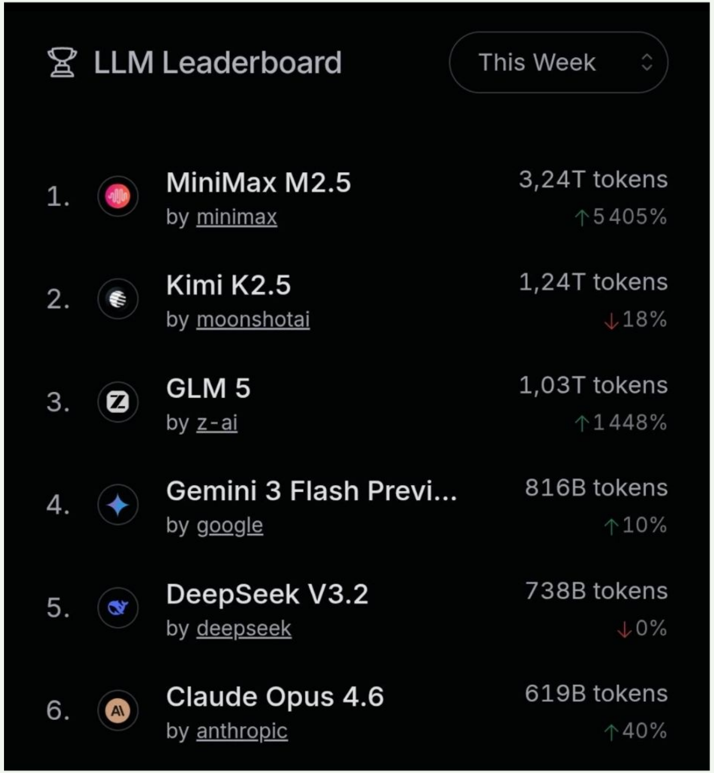
Top Models

Weekly usage of models across OpenRouter

February 16, 2026

Others	5.64T
MiniMax M2.5	2.57T
Kimi K2.5	1.04T
Gemini 3 Flash Preview	859B
GLM 5	803B
DeepSeek V3.2	745B
Grok 4.1 Fast	669B
Claude Opus 4.6	643B
Claude Sonnet 4.5	534B
Trinity Large Preview (free)	449B
Total	14T

OpenRouter Stats 2)



LLM Leaderboard

This Week

1.	MiniMax M2.5 by minimax	3,24T tokens ↑5 405%
2.	Kimi K2.5 by moonshotai	1,24T tokens ↓18%
3.	GLM 5 by z-ai	1,03T tokens ↑1 448%
4.	Gemini 3 Flash Previ... by google	816B tokens ↑10%
5.	DeepSeek V3.2 by deepseek	738B tokens ↓0%
6.	Claude Opus 4.6 by anthropic	619B tokens ↑40%

Let’s break down the top 3

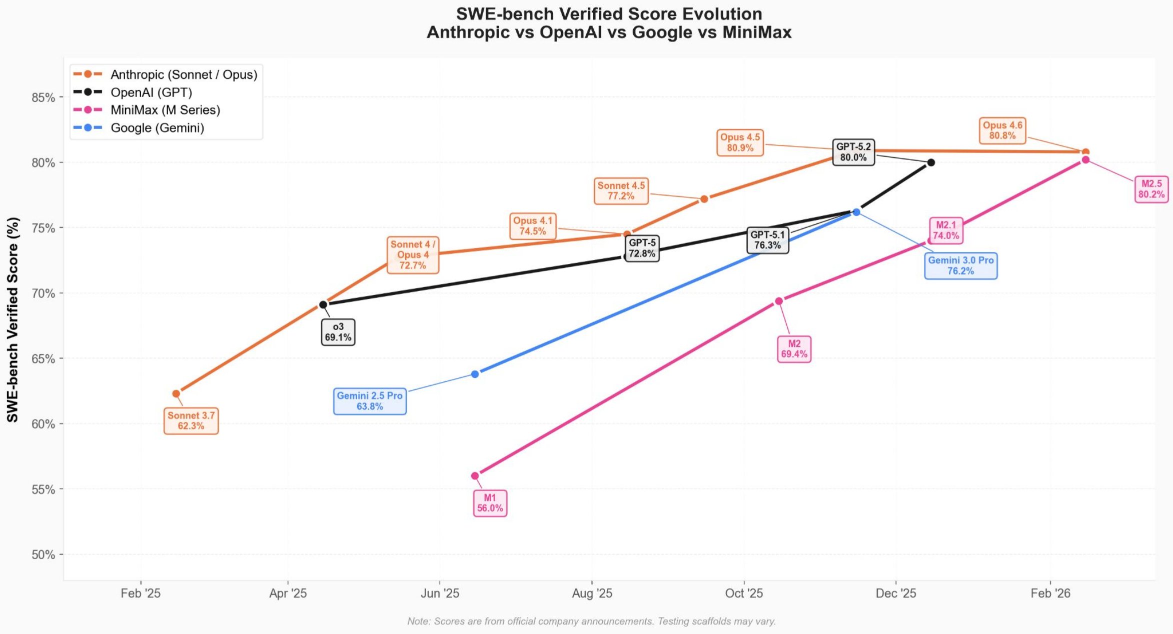
MiniMax M2.5 - small, fast, smart; good default choice

GLM 5 - very smart, but slow from high active parameter count

Kimi Thinking (and 2.5) - reasoning thus slow, best multimodal model

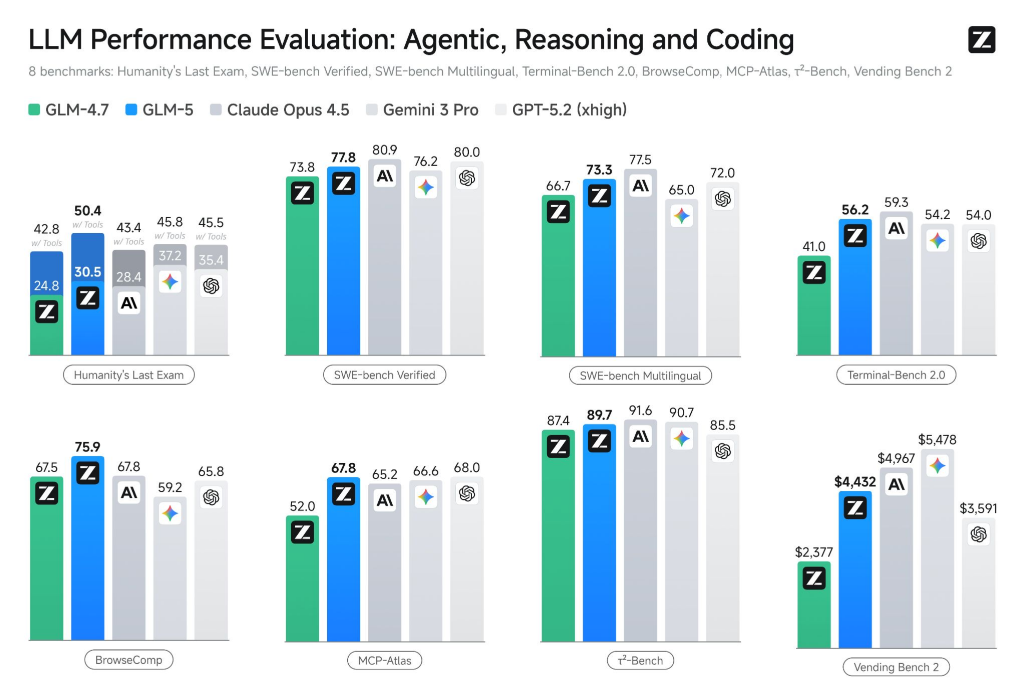
MiniMax M2.5

small, fast, smart; good default choice



GLM 5

very smart, but slow from high active parameter count



Kimi K2 Thinking Breakdown

Interesting Stats

- 1T parameter model
- Leader on humanity's’ last exam
- Tau-squared bench for agentic tool use
- 200-300 continuous tool calls

Architecture

- Sparse MoE architecture
- Bigger vocab compared with DeepSeek R1
- More experts, fewer attention heads per expert

Kimi K2 Thinking: Interleaved Reasoning

Model alternates between thinking and tool use

Compared to chain of thought, where all the reasoning is upfront, interleaved:

- Mimics humans – Reflect after each action, adjust approach dynamically
- Maintains coherence
- Solves complexity – Breaks ambiguous problems into adaptive subtasks through continuous feedback

Example:
Solved PhD-level geometry problem using 23 interleaved reasoning + tool call cycles

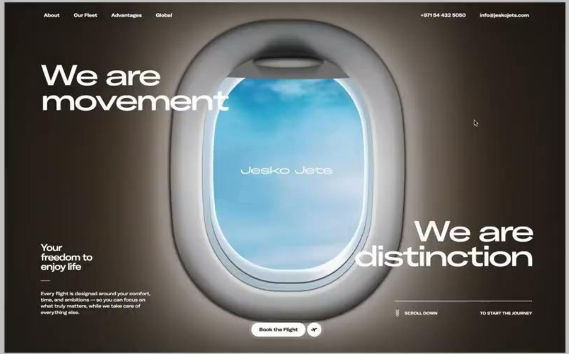
Kimi K2 Thinking: Tool Call Training

5-Step Training Pipeline:

- Gathered 3K+ real tools from GitHub
- Generated 20K+ synthetic tools via clustering
- Created synthetic agents with varied configurations
- Simulated diverse multi-turn usage scenarios
- Evaluated and filtered all trajectories for quality

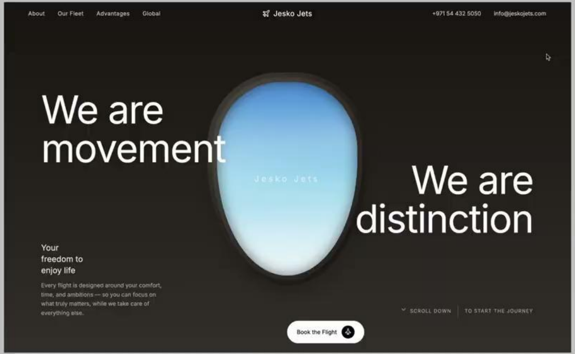
Kimi K2.5: coding with vision

Original Video (Cr: Jesko Jet)



Every flight is designed around your comfort, time, and ambitions — so you can focus on what truly matters, while we take care of everything else.

Kimi's Output

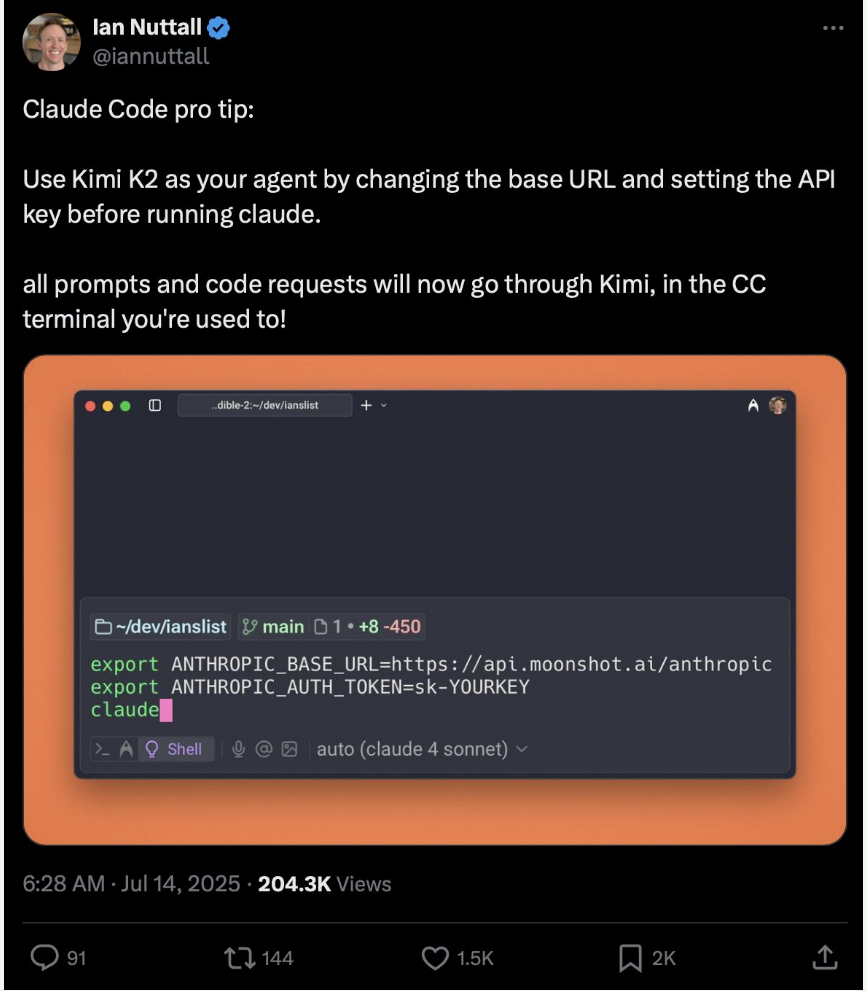


Every flight is designed around your comfort, time, and ambitions — so you can focus on what truly matters, while we take care of everything else.

My favorite dev workflows and harnesses

Claude Code CLI

(cheaper, faster, and just as smart)



Ian Nuttall
@iannuttall

Claude Code pro tip:

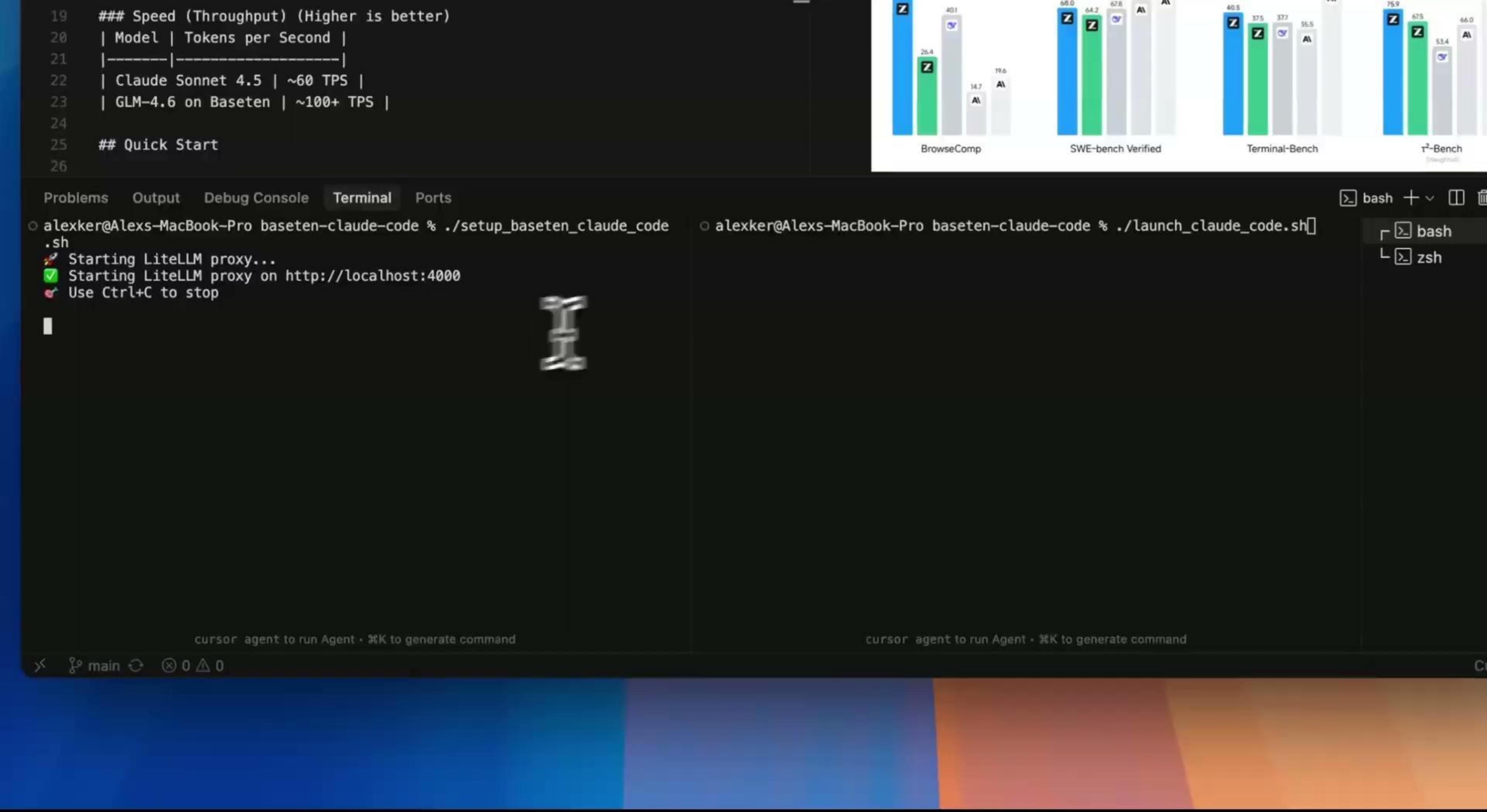
Use Kimi K2 as your agent by changing the base URL and setting the API key before running claude.

all prompts and code requests will now go through Kimi, in the CC terminal you're used to!

```
~/dev/ianslist  main  1 • +8 -450

export ANTHROPIC_BASE_URL=https://api.moonshot.ai/anthropic
export ANTHROPIC_AUTH_TOKEN=sk-YOURKEY
claude
```

6:28 AM · Jul 14, 2025 · **204.3K** Views



```
### Speed (Throughput) (Higher is better)
| Model | Tokens per Second |
|-------|-----------------|
| Claude Sonnet 4.5 | ~60 TPS |
| GLM-4.6 on Baseten | ~100+ TPS |

## Quick Start

```

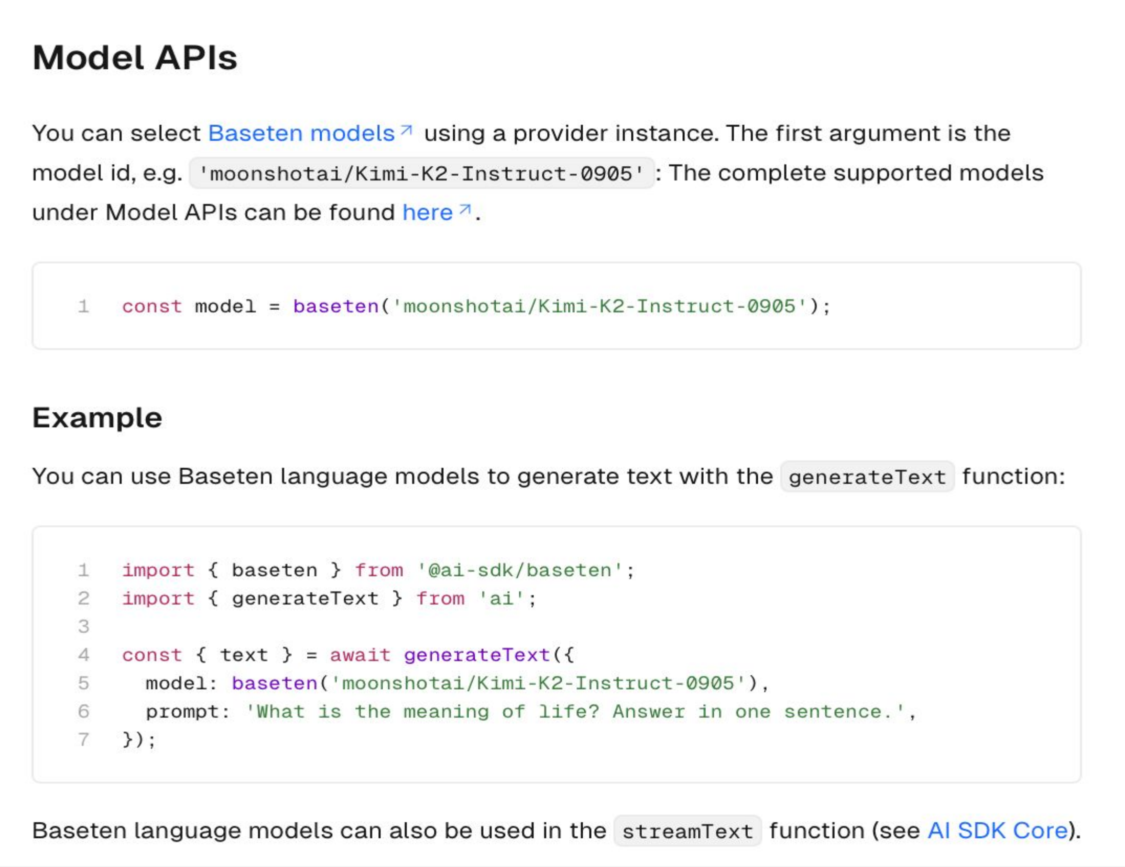
```
alexker@Alexs-MacBook-Pro baseten-claude-code % ./setup_baseten_claude_code
.sh
🚀 Starting LiteLLM proxy...
✅ Starting LiteLLM proxy on http://localhost:4000
🛑 Use Ctrl+C to stop
```

```
alexker@Alexs-MacBook-Pro baseten-claude-code % ./launch_claude_code.sh
```

Instead of hacking these CLIs, what if there was a way to build in open source native IDE?

Vercel AI SDK V5

- Instantiate provider objects that could be used directly in your Next.js application
- Compatibility across streaming, async, batch etc
- Patterns to help you build AI apps in your language of choice



Model APIs

You can select Baseten models ↗ using a provider instance. The first argument is the model id, e.g. `'moonshotai/Kimi-K2-Instruct-0905'`: The complete supported models under Model APIs can be found here ↗.

```
const model = baseten('moonshotai/Kimi-K2-Instruct-0905');
```

Example

You can use Baseten language models to generate text with the `generateText` function:

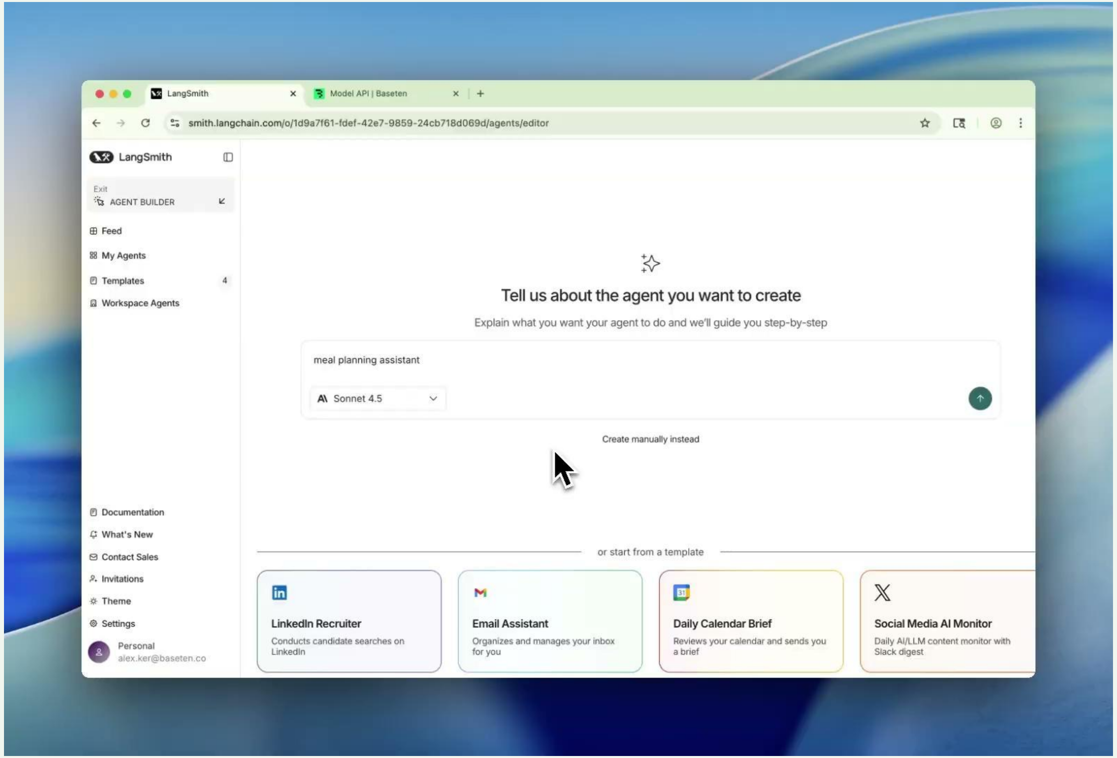
```
import { baseten } from '@ai-sdk/baseten';
import { generateText } from 'ai';

const { text } = await generateText({
  model: baseten('moonshotai/Kimi-K2-Instruct-0905'),
  prompt: 'What is the meaning of life? Answer in one sentence.',
});
```

Baseten language models can also be used in the `streamText` function (see AI SDK Core).

LangChain

- Agent Builder
- For knowledge workers, non technical
- Create an agent based on guided questions



Tell us about the agent you want to create

Explain what you want your agent to do and we'll guide you step-by-step

meal planning assistant

Sonnet 4.5

Create manually instead

or start from a template

LinkedIn Recruiter Conducts candidate searches on LinkedIn

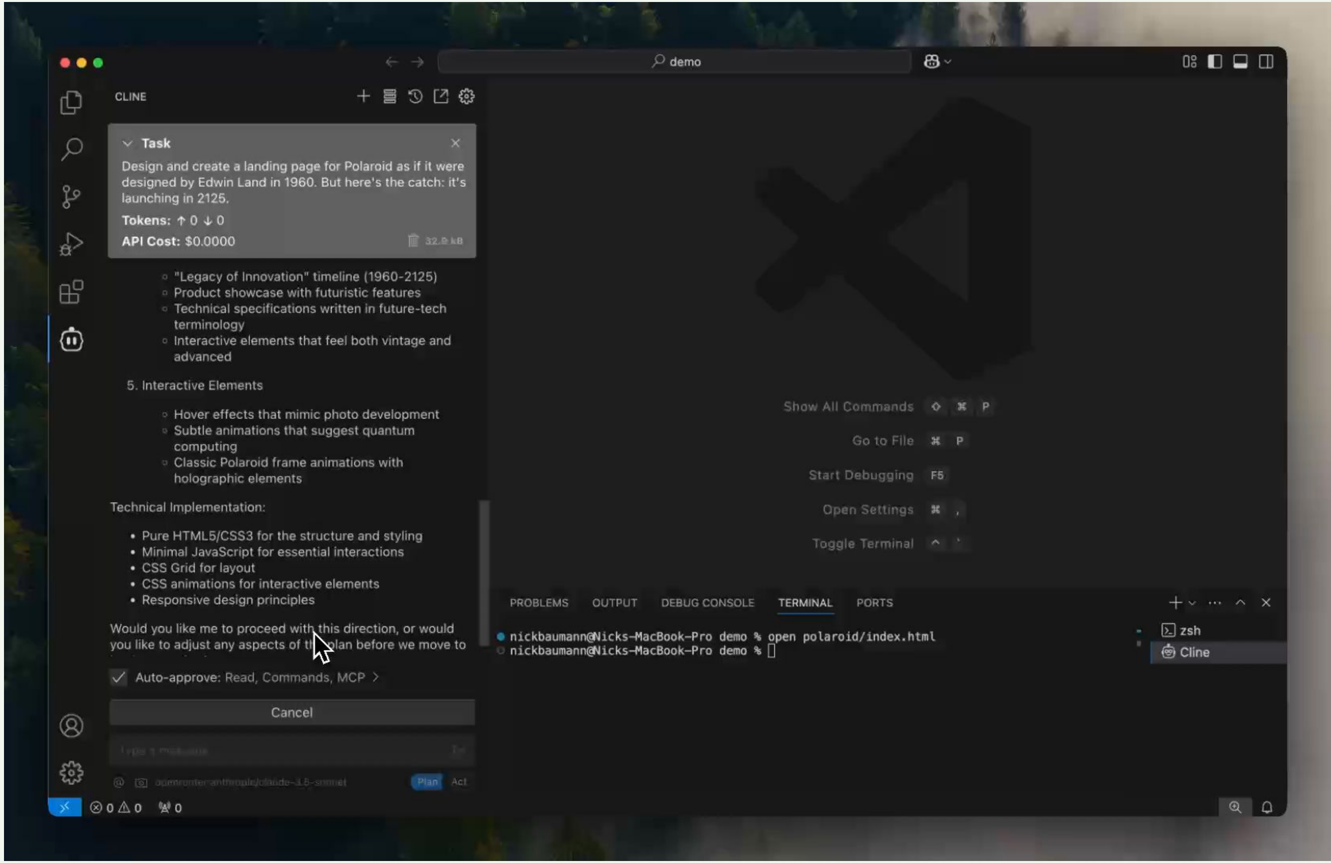
Email Assistant Organizes and manages your inbox for you

Daily Calendar Brief Reviews your calendar and sends you a brief

Social Media AI Monitor Daily AI/LLM content monitor with Slack digest

Cline

- Bring your own key
- Plan/Act modes
- Prewritten system prompts



Task

Design and create a landing page for Polaroid as if it were designed by Edwin Land in 1960. But here's the catch: it's launching in 2125.

Tokens: ↑ 0 ↓ 0

API Cost: $0.0000

- "Legacy of Innovation" timeline (1960-2125)
- Product showcase with futuristic features
- Technical specifications written in future-tech terminology
- Interactive elements that feel both vintage and advanced

5. Interactive Elements

- Hover effects that mimic photo development
- Subtle animations that suggest quantum computing
- Classic Polaroid frame animations with holographic elements

Technical Implementation:

- Pure HTML5/CSS3 for the structure and styling
- Minimal JavaScript for essential interactions
- CSS Grid for layout
- CSS animations for interactive elements
- Responsive design principles

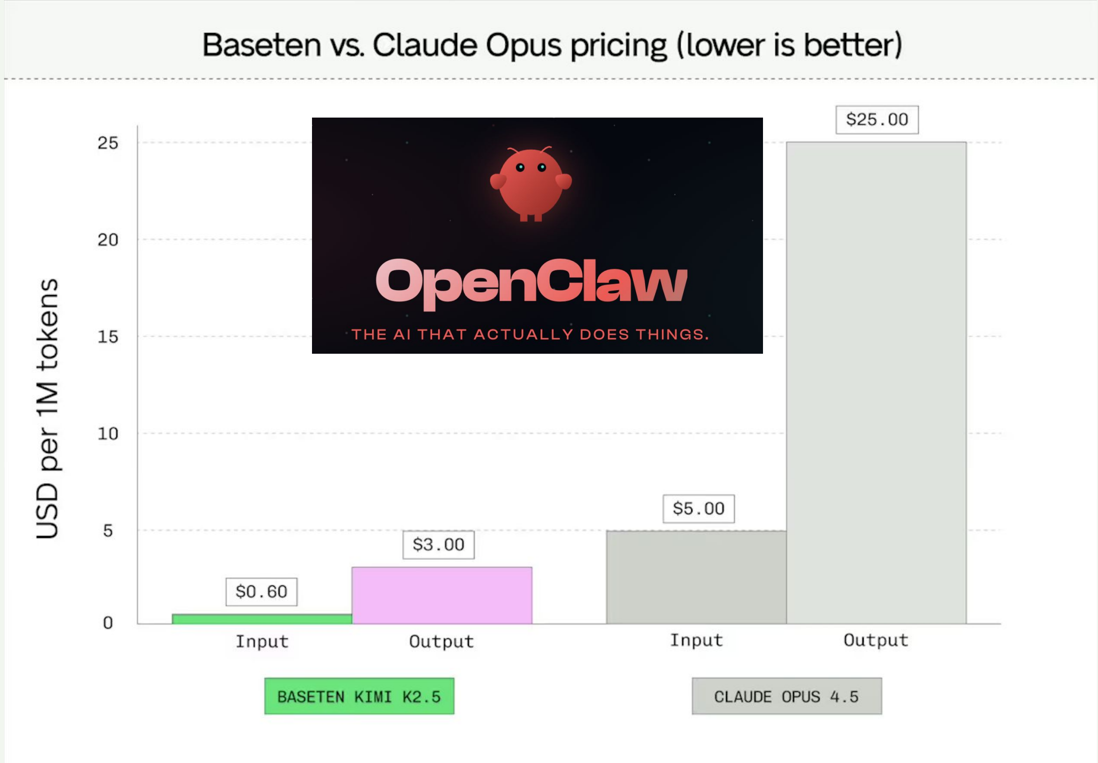
Would you like me to proceed with this direction, or would you like to adjust any aspects of the plan before we move to

☑ Auto-approve: Read, Commands, MCP >

```
nickbaumann@Nicks-MacBook-Pro demo % open polaroid/index.html
nickbaumann@Nicks-MacBook-Pro demo %
```

OpenClaw

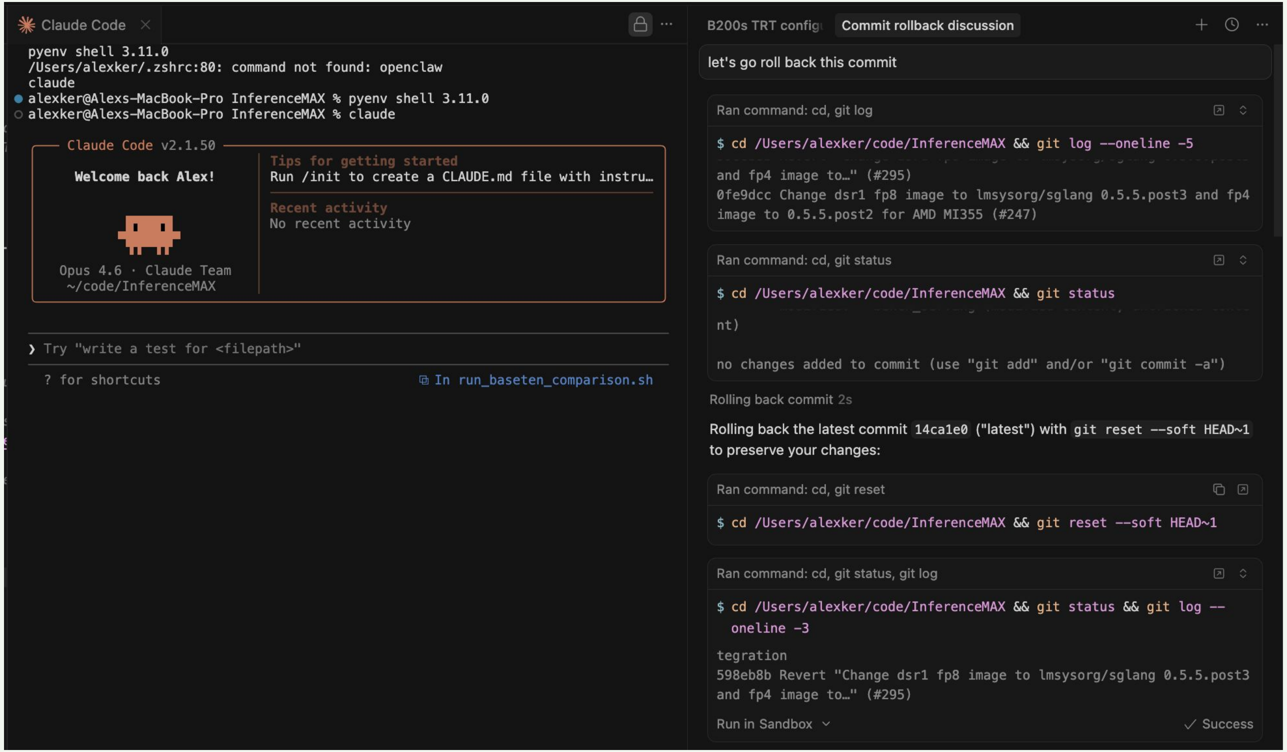
- Your own Jarvis, get started by choosing a model, and list of apps to be connected
- people have been able have tested crazy workflows
  - Booking restaurant reservations through voice call
  - Building apps while texting it feedback on whatsapp
- Kimi K2.5 is 8x cheaper and 4.5x faster than Claude Opus*



Baseten vs. Claude Opus pricing (lower is better)

Cursor

- my personal go-to setup
  - **Composer** (for faster, non-complex queries)
  - **Claude Code** (for planning projects, harder tasks that I need to be done correctly, not worried about time)



```
pyenv shell 3.11.0
/Users/alexker/.zshrc:80: command not found: openclaw
claude
● alexker@Alexs-MacBook-Pro InferenceMAX % pyenv shell 3.11.0
○ alexker@Alexs-MacBook-Pro InferenceMAX % claude

 Claude Code v2.1.50
        Welcome back Alex!        Tips for getting started
                                  Run /init to create a CLAUDE.md file with instru…

                                  Recent activity
                                  No recent activity

   Opus 4.6 · Claude Team
    ~/code/InferenceMAX

❯ Try "write a test for <filepath>"

  ? for shortcuts                                  ⧉ In run_baseten_comparison.sh
```

let's go roll back this commit

Ran command: cd, git log

```
$ cd /Users/alexker/code/InferenceMAX && git log --oneline -5
and fp4 image to…" (#295)
0fe9dcc Change dsr1 fp8 image to lmsysorg/sglang 0.5.5.post3 and fp4
image to 0.5.5.post2 for AMD MI355 (#247)
```

Ran command: cd, git status

```
$ cd /Users/alexker/code/InferenceMAX && git status
nt)

no changes added to commit (use "git add" and/or "git commit -a")
```

Rolling back commit 2s

Rolling back the latest commit `14ca1e0` ("latest") with `git reset --soft HEAD~1` to preserve your changes:

Ran command: cd, git reset

```
$ cd /Users/alexker/code/InferenceMAX && git reset --soft HEAD~1
```

Ran command: cd, git status, git log

```
$ cd /Users/alexker/code/InferenceMAX && git status && git log --
  oneline -3
tegration
598eb8b Revert "Change dsr1 fp8 image to lmsysorg/sglang 0.5.5.post3
and fp4 image to…" (#295)
```

Run in Sandbox ⌄ ✓ Success

Performant inference for coding models

Case study: **Autocomplete**

- Care about TTFT
- Long prefill, short decode
- Keeps up with user workflow

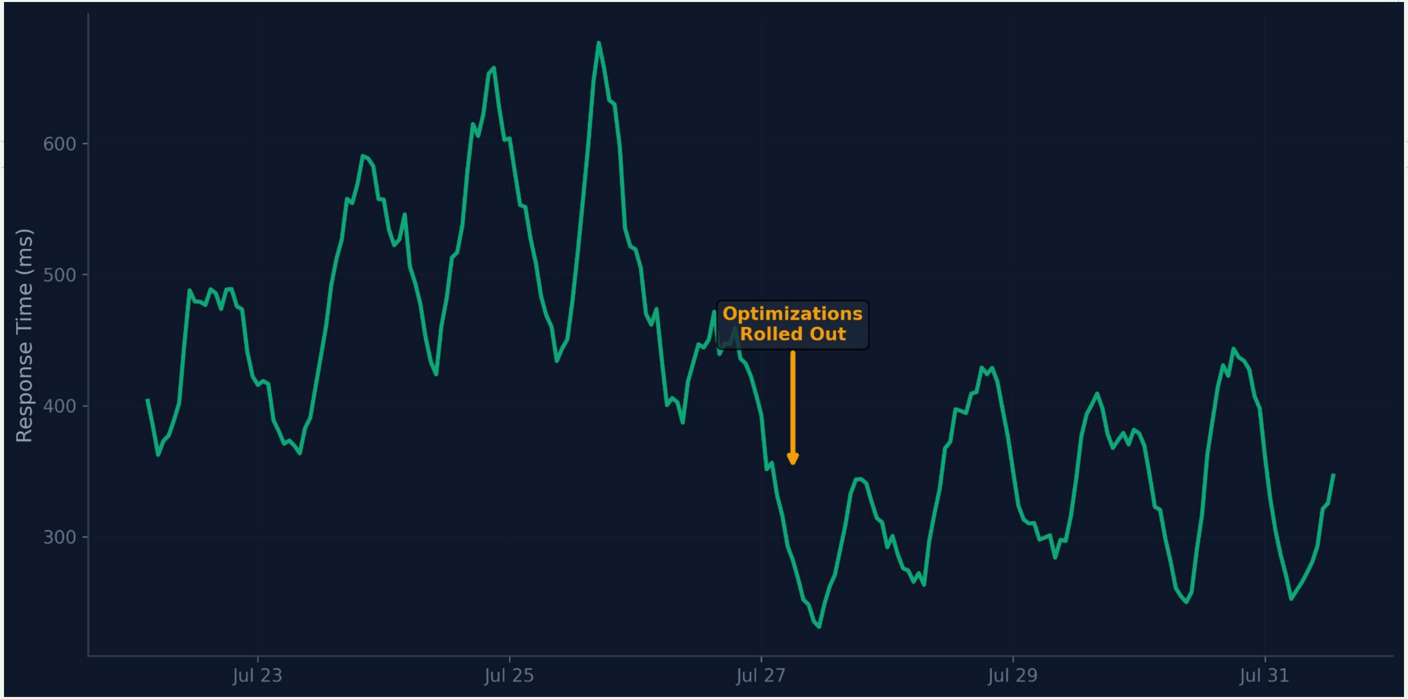
How?

KV cache re-use

KV-aware routing

N-gram speculation

Results!


Optimizations Rolled Out
Response Time (ms)
600
500
400
300
Jul 23
Jul 25
Jul 27
Jul 29
Jul 31

General Key Takeaways

1. If you're just using Claude Code or Codex, you're missing out
2. Open models have closed the gap
3. Developers have built a rich ecosystem of tools around them
4. The best thing is to **experiment** and not limit yourself
5. If you care about build delightful UX with AI, you should care about: **performance, reliability & control**

Actionable Steps

- **Delegate** and **review** like a staff engineer
- Develop your **own taste** of models & harnesses combinations. Spend time and let them do things for you!
- The models are very capable given the *right information*
    - The golden question: “Is there sufficient context for the agent to figure out what the answer is?”
    - This may solve most of your problems

Thank you!

instagram.com/thealexker

x.com/thealexker

linkedin.com/in/alex-ker