Running Claude Code for Free: Two Methods

Claude Code's terminal harness, your projects, CLAUDE.md files, MCP integrations, everything, works with any Anthropic-compatible API endpoint. That means you can swap out the paid Anthropic backend for either a local open-source model or a cloud-based free-tier router. Neither costs you a dollar.

By: Nate Herk

Method 1: Ollama (Local Models)

Best for: Privacy, offline use, no rate limits, zero ongoing cost
Trade-off: You're limited by your hardware; larger/smarter models need more RAM/VRAM. But there are some cloud models on Ollama that you can use but it will no longer be fully local and private.

Step 1: Install Ollama

Download from ollama.com for your OS (macOS, Windows, Linux). Once installed, Ollama runs silently in the background.

Step 2: Pick a Model Based on Your Hardware

Before pulling a model, ask an AI: *"I have [X GB RAM / GPU], what Ollama model sizes can I run?"* General guidance:

RAM / VRAM	Recommended Model Size
8 GB	7B–8B (quantized)
16 GB	14B–20B
32 GB	30B+ or full-precision smaller models
Apple Silicon (unified memory)	24B+ comfortable

Claude Code needs **at least 64K context**. Stick to models with 128K+ context windows.

Top recommended models (April 2026):

Model	Size	Context	Notes
`qwen3-coder`	30B	128K	Best overall for agentic coding
`devstral-2-small`	24B	128K	Multi-file agentic tasks
`granite3.3:8b`	8B	128K	Lightweight, strong tool calling
`glm-4.7-flash`	9B	128K	Fast, GPU-friendly

Step 3: Pull Your Model

```shell
ollama pull qwen3-coder
```

This downloads the model locally. One-time download, then it's yours.

Cloud shortcut: Ollama also offers cloud-hosted models (e.g. `glm-5:cloud`, `kimi-k2.5:cloud`) that don't require a local download, you just run them directly. Free tier is limited but works for getting started.

Step 4: Point Claude Code at Ollama

Option A, Easiest (Ollama v0.15+):

```shell
ollama launch claude
```

Ollama auto-configures all environment variables and launches Claude Code.

Option B, Manual (any version):

```shell
export ANTHROPIC_BASE_URL="http://localhost:11434"
export ANTHROPIC_AUTH_TOKEN="ollama"
export ANTHROPIC_API_KEY=""

claude --model qwen3-coder
```

Option C, Permanent via `~/.claude/settings.json`:

```json
{
  "env": {
    "ANTHROPIC_BASE_URL": "http://localhost:11434",
    "ANTHROPIC_AUTH_TOKEN": "ollama",
    "ANTHROPIC_API_KEY": ""
  }
}
```

Step 5: First-Time Login

When Claude Code prompts you to log in, select **"Anthropic Console account (API usage billing)"**. You don't need credits, this just enables the API flow that lets it talk to Ollama.

What You Lose vs. Anthropic Models

- **Native web search**, use an MCP-based search tool (Brave, Tavily) instead
- **Prompt caching**, each turn re-ingests full context (fine since cost is $0)
- **Raw capability**, open-source models are good but not Opus 4.6 tier

To Switch Back to Anthropic

```shell
unset ANTHROPIC_BASE_URL
unset ANTHROPIC_AUTH_TOKEN
unset ANTHROPIC_API_KEY
claude
```

Method 2: OpenRouter (Free Cloud Models)

Best for: Better model quality than local, no hardware requirements, cloud speed
Trade-off: Rate limits apply; requires an internet connection

Step 1: Set Up OpenRouter

1. Create an account at openrouter.ai
2. Generate an API key
3. **Add $10 in credits**, this unlocks 1,000 free-model requests/day (vs. 50 without). The $10 only gets consumed if you use paid models. Free (`:free`) models cost $0 regardless.

Account State	Free Model Requests/Day
No credits	50/day
$10+ credits	1,000/day

Step 2: Choose a Free Model

Look for models tagged `:free` on OpenRouter. Top picks (April 2026):

Model ID	Context	Notes
`qwen/qwen3-coder:free`	262K	Best overall
`mistralai/devstral-2:free`	256K	Multi-file agentic tasks
`moonshotai/kimi-k2.5:free`	256K	Strong general agent
`deepseek/deepseek-v3.1-nex-n1:free`	131K	Agent-optimized

Avoid `openrouter/auto:free`, it may route to models with poor tool calling.

Step 3: Configure All 6 Model Slots

Claude Code has 6 internal model slots. Every unset slot silently routes to a paid Anthropic model. Set them all.

Add to `~/.zshrc` / `~/.bashrc`:

```shell
export ANTHROPIC_BASE_URL="https://openrouter.ai/api"
export ANTHROPIC_AUTH_TOKEN="sk-or-your-key-here"
export ANTHROPIC_API_KEY=""

export ANTHROPIC_MODEL="qwen/qwen3-coder:free"
export ANTHROPIC_DEFAULT_SONNET_MODEL="qwen/qwen3-coder:free"
export ANTHROPIC_DEFAULT_OPUS_MODEL="qwen/qwen3-coder:free"
export ANTHROPIC_DEFAULT_HAIKU_MODEL="qwen/qwen3-coder:free"   #
← most common leak
export ANTHROPIC_SMALL_FAST_MODEL="qwen/qwen3-coder:free"
export CLAUDE_CODE_SUBAGENT_MODEL="qwen/qwen3-coder:free"

export CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC=1
```

Why Haiku matters: `ANTHROPIC_DEFAULT_HAIKU_MODEL` fires a background call on *every single prompt* to generate the session title. It's the most common billing leak.

Or set in `/.claude/settings.json`:

```json
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://openrouter.ai/api",
    "ANTHROPIC_AUTH_TOKEN": "sk-or-your-key-here",
    "ANTHROPIC_API_KEY": "",
    "ANTHROPIC_MODEL": "qwen/qwen3-coder:free",
    "ANTHROPIC_DEFAULT_SONNET_MODEL": "qwen/qwen3-coder:free",
    "ANTHROPIC_DEFAULT_OPUS_MODEL": "qwen/qwen3-coder:free",
    "ANTHROPIC_DEFAULT_HAIKU_MODEL": "qwen/qwen3-coder:free",
    "ANTHROPIC_SMALL_FAST_MODEL": "qwen/qwen3-coder:free",
    "CLAUDE_CODE_SUBAGENT_MODEL": "qwen/qwen3-coder:free",
    "CLAUDE_CODE_DISABLE_NONESSENTIAL_TRAFFIC": "1",
    "CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS": "1"
  }
}
```

Step 4: Watch for Beta Header Errors

Claude Code sends Anthropic-specific `anthropic-beta` headers that non-Anthropic models don't understand, causing 400 errors. `CLAUDE_CODE_DISABLE_EXPERIMENTAL_BETAS=1` is supposed to strip these, but it has known bugs. Fallbacks:

- Pin an older stable version: `npm install -g @anthropic-ai/claude-code@2.1.68`
- Add `DISABLE_PROMPT_CACHING=1` to strip cache-related headers
- OpenRouter handles most of these automatically for supported models

Verify It's Working

Check your OpenRouter activity dashboard. A clean setup shows:

- ✅ All requests hitting your chosen `:free` model
- ✅ `usage: 0` (or $0.00) on all calls
- ✅ No calls to `anthropic/claude-*`

Which Method Should You Use?

	Ollama (Local)	OpenRouter (Cloud Free)
Cost	$0 forever	$0 (free models)
Hardware required	Yes	No
Privacy	100% local	Data sent to cloud
Model quality	Good (hardware-limited)	Good (not Anthropic tier)
Rate limits	None	1,000 req/day w/ $10
Internet required	No	Yes
Setup complexity	Low (one command)	Medium (6 env vars)

Use Ollama if you have a capable machine (16GB+ RAM) and care about privacy or offline access.
Use OpenRouter if you want cloud-quality models without hardware investment.

Context Management Tips (Both Methods)

Claude Code sends 40K–80K tokens per turn before you type anything. Keep sessions lean:

- Set `"autoCompactThreshold": 75` in settings.json (compacts at 75% vs. default 95%)
- Run `/compact` manually before starting a major new task
- Keep CLAUDE.md files under 2,000 tokens, they're injected into every request

---

Is This Allowed?

Yes. Using Claude Code with your own API key or third-party providers via `ANTHROPIC_BASE_URL` is explicitly supported. The ToS restriction (added early 2026) only applies to using a Claude Pro/Max OAuth token in third-party tools, not to API key usage.

Want to connect with others building and monetizing AI automation?

Become an AIS Plus Member