CLAUDE CODE + OLLAMA + GEMMA 4

Run Claude Code Locally for Free — Complete Setup Guide

Prerequisites

Before starting, make sure your system meets these requirements:

Requirement	Minimum	Notes
RAM	8 GB	16 GB recommended for smooth performance
Storage	10 GB free	Model files range from 3 GB to 18 GB
OS	macOS / Windows / Linux	All fully supported by Ollama
Node.js	v18 or later	Required to install Claude Code via npm
Internet	Required once	Only for initial download; fully offline after

Note

Node.js is required to install Claude Code. Download it from https://nodejs.org — get the LTS version. Run node --version in your terminal to verify.

1 Install Ollama — The AI Model Engine

Ollama is the runtime that downloads and runs AI models locally on your machine. It handles all the heavy lifting so you don't need to manage model weights manually.

Mac

- Go to https://ollama.com/download
- Click Download for Mac

- Open the downloaded file and install it like any normal app (no Terminal needed)

Windows

- Go to https://ollama.com/download
- Click Download and run the installer
- Follow the on-screen instructions

Linux

Run this single command in your terminal:

```
curl -fsSL https://ollama.com/install.sh | sh
```

Verify the Installation

After installing, open a terminal and run:

```
ollama --version
```

You should see output like: ollama version is 0.15.x

> ⚠ **Important**
>
> If you see 'unknown command: launch' errors later, your Ollama version is too old. Update Ollama by restarting it and selecting 'Restart to update' from the menu bar icon, or re-run the installer.

2 Download Gemma 4 — Choose Your Model Size

Gemma 4 is Google's open-source model family. Pick the variant based on how much RAM your machine has. This is a one-time download.

Model	RAM Needed	Download Size	Best For
`gemma4:e2b`	8 GB	~3 GB	Low-end machines, quick tasks
`gemma4:e4b`	16 GB	~7 GB	Recommended — best balance of speed and quality
`gemma4:26b`	32 GB	~18 GB	High-end machines, complex reasoning tasks

Run the pull command for your chosen model:

```
# Low-end (8 GB RAM)
ollama pull gemma4:e2b

# Recommended (16 GB RAM)
ollama pull gemma4:e4b

# High-end (32 GB RAM)
ollama pull gemma4:26b
```

This will show a download progress bar. Grab a coffee — it can take a few minutes depending on your internet speed.

Verify the Download

Once the download completes, confirm the model is ready:

```
ollama list
```

You should see your gemma4 model listed with its size and modification date.

> **Note**
>
> Optional quick test: run ollama run gemma4:e4b and type a simple question. If it responds, the model is working correctly. Type /bye to exit.

3 Install Claude Code

Claude Code is Anthropic's terminal-based coding agent. It reads your files, edits code, runs shell commands, and handles multi-step tasks from a conversational interface. You install it via npm.

Open your terminal and run:

```
# macOS and Linux
curl -fsSL https://claude.ai/install.sh | bash


# Windows (CMD)
curl -fsSL https://claude.ai/install.cmd -o install.cmd && install.cmd &&
del install.cmd
```

Verify Claude Code is Installed

```
claude --version
```

If you see a version number, you're good to go. If you see 'claude is not installed', re-run the install command above.

> **Note**
>
> You can also visit https://claude.com/product/claude-code to learn more about Claude Code features and find the latest install instructions.

4 Launch Claude Code with Gemma 4

Now connect everything together. Navigate to your project folder in the terminal, then launch Claude Code backed by your local Ollama model.

Navigate to Your Project

```
cd /path/to/your/project
```

Launch with Ollama

Use the `ollama launch` command — it automatically handles all the API routing and environment variables for you:

```
# Replace the model tag with whichever you downloaded

ollama launch claude --model gemma4:e2b

ollama launch claude --model gemma4:e4b

ollama launch claude --model gemma4:26b
```

Claude Code will start, ask permission to access your project folder (type yes), and drop you into an interactive session.

Alternative: Manual Environment Variable Setup

If `ollama launch` doesn't work on your system, set the environment variables manually and then run claude directly:

```
# macOS / Linux — add these to your ~/.zshrc or ~/.bashrc for persistence
export ANTHROPIC_AUTH_TOKEN=ollama
export ANTHROPIC_API_KEY=""
export ANTHROPIC_BASE_URL=http://localhost:11434

# Then launch Claude Code
claude --model gemma4:e4b
```

> ⚠ **Important**
>
> Claude Code routes different task types (quick tasks, standard work, complex reasoning) to different model tiers internally. When using Ollama, set these extra variables to ensure all requests stay local and don't accidentally call Anthropic's servers:
>
> export ANTHROPIC_DEFAULT_HAIKU_MODEL=gemma4:e4b
>
> export ANTHROPIC_DEFAULT_SONNET_MODEL=gemma4:e4b
>
> export ANTHROPIC_DEFAULT_OPUS_MODEL=gemma4:e4b

Using Claude Code — Quick Reference

Once inside a session, type your requests in plain English. Some examples to get started:

```
> Explain what this codebase does
> Write a Python function to parse JSON from a file
> Find all TODO comments in this project and list them
> Refactor this function to use async/await
> Add error handling to the API call in app.py
```

Shortcut / Command	What it does
`Tab`	Switch between planning and building mode
`Ctrl + P`	Open options panel (switch model, share session)
`Ctrl + C`	Exit Claude Code
`/loop`	Run a prompt on a recurring schedule (great for automation)
`--yes flag`	Skip confirmation prompts — useful for scripting and CI pipelines

Troubleshooting

'unknown command: launch' error

Your Ollama version is too old. Update it: on Mac/Windows, click the Ollama icon in the menu bar and select Restart to update. On Linux, re-run the install script.

Model is extremely slow or timing out

Gemma 4 requires significant RAM. If responses take longer than 2 to 3 minutes, you are likely running a model too large for your hardware. Drop down one size (e.g., from e4b to e2b).

Tool calls fail or Claude Code is stuck in a loop

Claude Code needs large context windows (64K+ tokens minimum) for agentic tasks. Gemma 4 has moderate tool-calling support. If you notice issues, consider creating a custom Modelfile with an expanded context window:

```
# Create a Modelfile
cat > Modelfile << 'EOF'
FROM gemma4:e4b
PARAMETER num_ctx 65536
EOF

# Build the custom model
ollama create gemma4-e4b-64k -f Modelfile

# Use it with Claude Code
ollama launch claude --model gemma4-e4b-64k
```

Ollama server is not running

If you get a connection error, start the Ollama server manually:

```
ollama serve
```

Then open a new terminal tab and run your ollama launch claude command.

Claude Code is calling Anthropic's servers instead of local Ollama

This happens when the role-model environment variables aren't set. Add all three of these to your shell profile (~/.zshrc on Mac or ~/.bashrc on Linux):

```
export ANTHROPIC_DEFAULT_HAIKU_MODEL=gemma4:e4b
export ANTHROPIC_DEFAULT_SONNET_MODEL=gemma4:e4b
export ANTHROPIC_DEFAULT_OPUS_MODEL=gemma4:e4b
```

> **Note**
>
> The `ollama launch` command handles all environment variables automatically when it works correctly. The manual variable setup is only needed as a fallback or for persistent shell sessions.

Key Points

- Everything runs on your machine — your code and prompts never leave localhost
- This setup is completely free: no Anthropic API costs, no rate limits
- Gemma 4 works well for most coding tasks; for complex agentic work, GLM-4.7-Flash or Qwen models may offer better tool-calling reliability
- Keep Ollama updated — the `ollama launch` command and Anthropic API compatibility are recent additions and improve with each release
- Claude Code's file editing, command execution, and multi-step behavior are handled by the CLI itself — they don't depend on Anthropic's infrastructure

> **Note**
>
> For the latest Claude Code documentation and updates, visit:
>
> https://claude.com/product/claude-code
>
> https://docs.ollama.com/integrations/claude-code