BUILD YOUR OWN 24/7 AI ASSISTANT

Hermes Agent

The Complete Setup & Automation Blueprint

Install on any Linux VPS

Connect to WeChat & Telegram

Configure 5 AI Providers

Automate 10+ Daily Tasks

No Coding Required

Table of Contents

Chapter 1: What You're Building — Your Personal AI Agent

> **"An AI agent you actually own, running on your own server, connected to your phone, working 24/7."**

Before we dive into installation, let's get crystal clear on what exactly you're building and why it's worth your time.

1.1 Hermes Agent vs. ChatGPT vs. Claude

Most people interact with AI through a web browser. You type something in, get an answer, close the tab. Here's what that looks like:

	ChatGPT / Claude (Web)	
Runs where?	Someone else's cloud	**Your own server**
Works when you're offline?	❌	✅ Runs 24/7
Talks to you on phone?	Web app only	**WeChat, Telegram, Discord**
Remembers you between chats?	Limited	**Permanent memory**
Can run commands on your server?	❌	✅ Full shell access
Cost	$20/mo + API usage	**$5/mo VPS + pennies in API**

The key difference: Hermes Agent is an *autonomous agent*, not just a chatbot. It can:

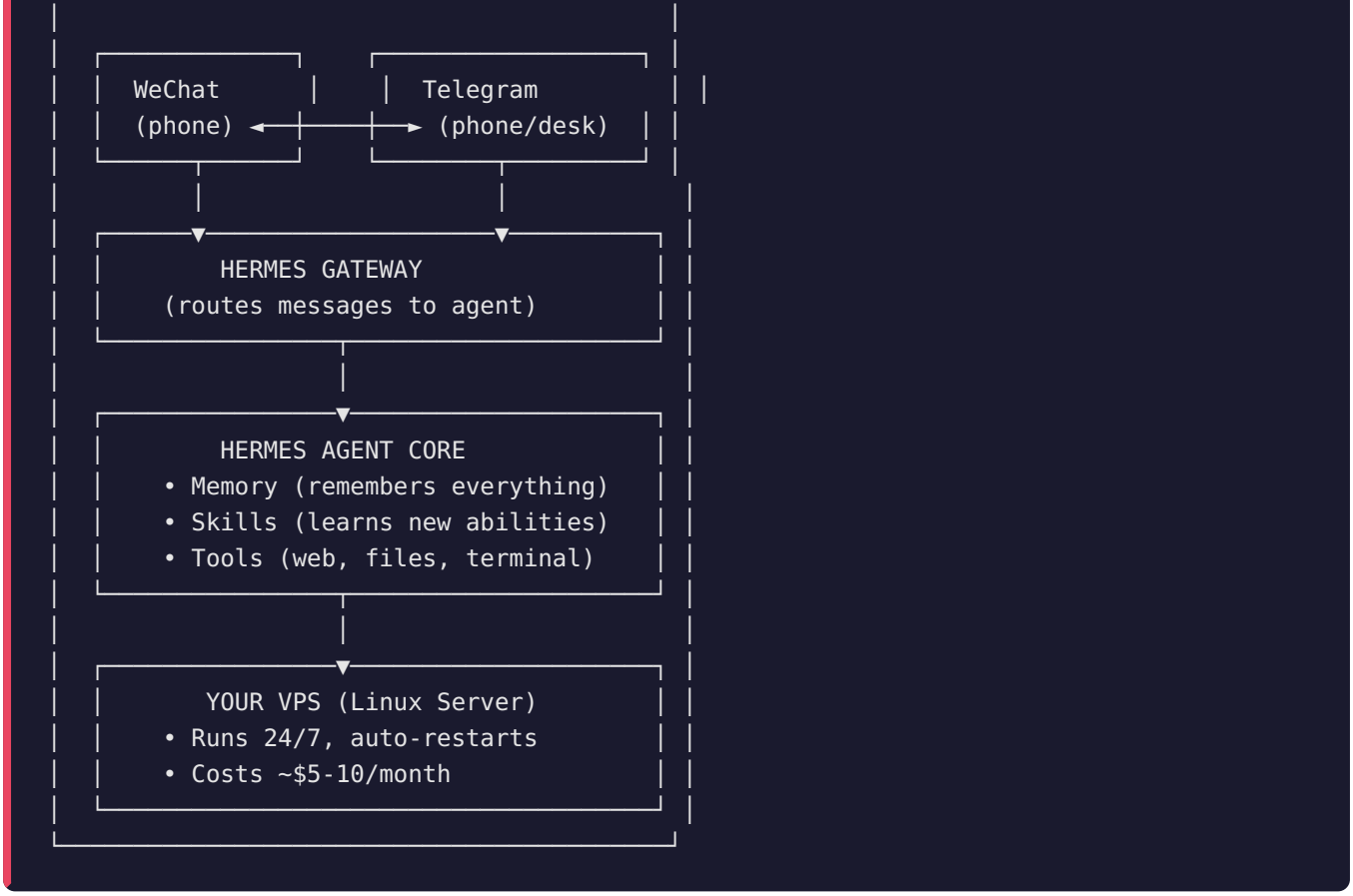
- **Schedule and execute tasks** — send you a daily news briefing every morning at 8 AM
- **Monitor things** — watch a website for changes, check stock prices, alert you
- **Run code** — write Python scripts, analyze data, manage files
- **Learn new skills** — teach it a workflow once, it remembers forever
- **Connect to everything** — email, calendar, smart home, GitHub, databases

Think of it as having a personal assistant that never sleeps, never takes a day off, and costs less than a Netflix subscription.

1.2 What You'll Have at the End of This Guide

By the time you finish this book, you will have:

```
┌──────────────────────────────────────┐
│          YOUR HERMES AGENT           │
```



1.3 What Makes Hermes Agent Special?

Skills System

This is Hermes' superpower. When you teach your agent how to do something — or when it figures something out on its own — it saves that knowledge as a **skill**. Next time you need the same thing, it already knows how.

Example: You ask your agent to "check if any new GitHub releases mention my company name." The agent figures out the API calls, the search logic, the notification format. It saves this as a skill called `github-mention-watcher`. Tomorrow, you just say "run the watcher" and it works — no re-explaining needed.

Persistent Memory

Hermes remembers who you are. Your preferences. Your environment. Things you taught it last week. It stores this in a memory system that survives restarts, reboots, and even server migrations.

Multi-Platform

You talk to your agent through **WeChat** (from your phone), **Telegram** (from your desktop), and **CLI** (directly on the server). The same agent, the same memory, the same skills — different entry points.

Provider Flexibility

You're not locked into any AI company. Use **OpenAI** today, switch to **Anthropic** tomorrow, use **Google Gemini** for free, fall back to **DeepSeek** when you're on a budget. The agent sees them all the same way.

1.4 Real Things People Do With Hermes

Category	Example
Daily briefings	"Every morning at 8 AM, send me weather + top 3 news + my calendar"
Website monitoring	"Watch product X on Amazon and alert me when price drops below $50"
Research assistant	"Research competitors in my space and write a summary to this file"
Content creation	"Write a blog post from these notes and publish to WordPress"
Financial tracking	"Check my stock portfolio every Friday and send a summary"
DevOps helper	"Check disk usage, restart the web service if memory is high"
Language learning	"Send me 5 Japanese flashcards every evening"
Email management	"Summarize my inbox, flag urgent emails"

1.5 The Cost Breakdown

Here's what this will actually cost you per month:

Item	Cost	Notes
VPS (Lighthouse / GCP / DigitalOcean)	**$5-10/mo**	Smallest tier is plenty
AI API usage (typical)	**$2-10/mo**	Gemini Flash is nearly free
Total	**~$7-20/mo**	Less than ChatGPT Plus

If you're smart about which model you use (Gemini Flash for simple tasks, Claude for complex ones), you can keep this under **$10/month total**.

1.6 What You Need to Start

Before Chapter 2, make sure you have:

- [ ] A Linux server (VPS) — we'll cover this in Chapter 5
- [ ] An email address (for API account signups)
- [ ] A phone with WeChat or Telegram installed
- [ ] A credit/debit card (for the VPS — costs $5-10/mo)

That's it. No coding experience needed. If you can copy and paste commands, you can build this.

Ready to build? In Chapter 2, we'll do a one-command install and have your first conversation with Hermes.

Chapter 2: One-Command Install — Your First Hermes Conversation

> **"No complicated setup. No missing dependencies. One command, and you're talking to your AI."**

Hermes Agent was designed to be installable with a single curl command. In this chapter, we'll get it running on your Linux server and have your first conversation — all in under 10 minutes.

2.1 Prerequisites

You need: - A Linux server (Ubuntu 22.04 / Debian 12 recommended) - Root or sudo access - At least 1 GB RAM, 10 GB disk (the smallest VPS tier works) - Python 3.10+ (most modern Linux distros have this)

> **Don't have a server yet?** Skip to Chapter 5 which covers setting up Tencent Cloud Lighthouse, GCP, and other VPS options. Then come back here.

2.2 The One-Command Install

SSH into your server, then run:

```
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | ba
```

What this does: 1. Detects your OS and architecture 2. Creates a Python virtual environment at `~/.hermes/hermes-agent/venv/` 3. Installs Hermes Agent and all Python dependencies 4. Sets up the `hermes` command in your PATH

When it finishes, you'll see something like:

```
✅ Hermes Agent installed successfully!
Run 'hermes' to start chatting.
```

If the command isn't found after install

The installer adds Hermes to your shell profile, but it won't take effect in the current session. Run this to activate it:

```
source ~/.bashrc
```

Or log out and SSH back in.

> ⚠️ **Common pitfall:** Some VPS providers ship minimal images without `curl`. If you get "command not found: curl", install it first:
>
> `bash sudo apt update && sudo apt install curl -y # Ubuntu/Debian`
>
> For CentOS/RHEL: `sudo yum install curl -y`

2.3 Your First Conversation

Just type:

```
hermes
```

You'll see a welcome banner, and Hermes will verify your environment. Then type anything you want.

```
┌──────────────────────────────────────────┐
│           Hermes Agent v0.x.x            │
│     Your autonomous AI assistant         │
└──────────────────────────────────────────┘
Model: Not configured
Provider: Not configured
Type '/help' for available commands.

You: Hello! What can you do?
```

If it asks you to configure a model first (because no API key is set), that's fine — we'll do that in the next chapter. But if you already have an API key set in your environment, Hermes will start working immediately.

Quick test queries to try:

```
What's the current date and time?
Show me disk space on this server
List the files in my home directory
What's my IP address?
```

2.4 Exiting Hermes

```
/quit
```

Or press `Ctrl+C` / `Ctrl+D`.

2.5 Understanding What Just Installed

Here's what's now on your server:

```
~/.hermes/                          # Hermes home directory
├── config.yaml                     # Your configuration
├── .env                            # API keys (you'll create this)
```

```
├── hermes-agent/          # Source code
│   ├── venv/              # Python virtual environment
│   └── ...                # The agent code
├── skills/                # Installed skills
├── sessions/              # Chat history
└── logs/                  # Error logs
```

Key paths to remember: - `~/.hermes/config.yaml` — main configuration file - `~/.hermes/.env` — API keys and secrets (never share this!) - `~/.hermes/sessions/` — all your conversations are saved here

2.6 Updating Hermes

Updates come out regularly with new features and fixes:

```
hermes update
```

Or to check your current version:

```
hermes --version
```

2.7 Uninstalling (Just in Case)

If you ever need to remove it:

```
hermes uninstall
```

This removes the `~/.hermes/` directory and the `hermes` command.

You just installed an AI agent on your own server. That's the hardest part — everything from here is configuration and customization.

In **Chapter 3**, we'll connect Hermes to an AI provider so it can actually think and respond. You'll learn how to set up OpenAI, Anthropic, Google Gemini, OpenRouter, and even use them together with automatic fallback.

Chapter 3: Multiple Provider Setup — Pick Your Brain

> **"The best part of Hermes: you're not locked into any AI company. Switch models like changing TV channels."**

Hermes works with over 20 AI providers. You can use any of them — or several at once with automatic fallback. This chapter covers the most popular ones step by step.

3.1 How Providers Work in Hermes

You need two things:

1. **An API key** — from the AI provider (OpenAI, Anthropic, Google, etc.)
2. **Configuration** — tell Hermes which provider and model to use

All API keys go in `~/.hermes/.env`. All settings go in `~/.hermes/config.yaml` (or use the interactive model picker).

> **Security rule:** Never put API keys directly in `config.yaml`. Always use the `.env` file. The `.env` file should never be shared or committed to GitHub.

3.2 Option A: Google Gemini (Best Free Option)

Why choose this: Gemini 2.5 Flash is extremely capable and has a generous free tier. I use this as my primary provider.

Step 1: Get an API key

1. Go to aistudio.google.com/apikey
2. Click **"Create API Key"**
3. Copy the key (it starts with `AIza...`)

Step 2: Add to your server

SSH in and run:

```
echo 'GOOGLE_API_KEY=*** >> ~/.hermes/.env
```

Replace `your_key_here` with the actual key you copied.

Step 3: Configure Hermes

Run the interactive model picker:

```
hermes model
```

Navigate to **Google**, select **Gemini 2.5 Flash**, and confirm.

Or set it directly:

```
hermes config set model.provider google
hermes config set model.default gemini-2.5-flash
```

Verify it works:

```
hermes chat -q "Hello! What model are you using?"
```

It should respond and mention Gemini.

3.3 Option B: OpenRouter (One Key for 200+ Models)

Why choose this: One API key gives you access to every major model — Claude, GPT-4, Gemini, DeepSeek, Llama, Mistral, and many more. You pay per-use, per-model.

Step 1: Get an API key

1. Go to openrouter.ai/keys
2. Sign up and create a key
3. Add $5-10 credit (lasts months for typical use)

Step 2: Add to `.env`

```
echo 'OPENROUTER_API_KEY=*** >> ~/.hermes/.env
```

Step 3: Configure

```
hermes config set model.provider openrouter
hermes config set model.default openai/gpt-4o-mini
```

Available models include: - `openai/gpt-4o-mini` — cheap and fast (~$0.15/million tokens) - `openai/gpt-4o` — most capable OpenAI model - `anthropic/claude-sonnet-4` — best for coding - `anthropic/claude-haiku-4` — fast and cheap - `google/gemini-2.5-flash` — free tier available - `deepseek/deepseek-chat` — very cheap, good quality - `meta-llama/llama-3-8b` — open-source, runs fast

3.4 Option C: Anthropic Claude (Best for Coding)

Why choose this: Claude Sonnet 4 is arguably the best model for programming and complex reasoning tasks.

Step 1: Get an API key

1. Go to console.anthropic.com
2. Sign up and generate an API key
3. Add credit ($5-20 depending on usage)

Step 2: Add to `.env`

```
echo 'ANTHROPIC_API_KEY=*** >> ~/.hermes/.env
```

Step 3: Configure

```
hermes config set model.provider anthropic
hermes config set model.default claude-sonnet-4
```

3.5 Option D: OpenAI (GPT-4)

Step 1: Get an API key

1. Go to platform.openai.com/api-keys
2. Create a new key
3. Add credit (requires a paid account)

Step 2: Add to `.env`

```
echo 'OPENAI_API_KEY=*** >> ~/.hermes/.env
```

Step 3: Configure

```
hermes config set model.provider openai
hermes config set model.default gpt-4o
```

3.6 Option E: Custom / Local Models

If you're running a local LLM (via Ollama, llama.cpp, vLLM, etc.), point Hermes to it:

```
hermes config set model.provider custom
hermes config set model.base_url http://localhost:11434/v1  # Ollama example
hermes config set model.default llama3.1
```

3.7 Advanced: Multi-Provider Fallback

This is where Hermes really shines. You can set up a **primary model** and **fallback models** — if the primary fails (rate limit, outage, etc.), Hermes automatically tries the next one.

Edit `~/.hermes/config.yaml` and set:

```
model:
  provider: openrouter
  default: openai/gpt-4o-mini
  fallbacks:
    - provider: google
      model: gemini-2.5-flash
    - provider: anthropic
      model: claude-haiku-4
```

Now your agent tries GPT-4o-mini first. If OpenRouter is down, it switches to Gemini. If Gemini is unavailable, it falls back to Claude. Your agent never goes silent.

3.8 Cost Comparison

Model	Cost per 1M input tokens	Good for
Gemini 2.5 Flash	Free tier → $0.15	Everyday tasks, fast response
Claude Haiku 4	~$0.80	Quick answers, classification
GPT-4o mini	~$0.15	General use, very cheap
DeepSeek Chat	~$0.27	Coding, analysis
Claude Sonnet 4	~$3.00	Complex coding, deep reasoning
GPT-4o	~$2.50	General purpose premium
Claude Opus 4	~$15.00	Maximum capability

My recommendation for you: Start with **Gemini 2.5 Flash** (free tier covers most needs) and add **OpenRouter** for when you need Claude's coding abilities.

3.9 Quick Config Cheat Sheet

```
# Set provider
hermes config set model.provider openrouter

# Set default model
hermes config set model.default openai/gpt-4o-mini
```

```
# Set environment variable (replace KEY with actual)
echo 'PROVIDER_API_KEY=*** >> ~/.hermes/.env

# Verify everything
hermes doctor

# Test
hermes chat -q "Hello! Who are you?"
```

In Chapter 4, we'll connect Hermes to WeChat so you can talk to your agent from your phone. This is where it gets really fun — your personal AI assistant in your pocket, 24/7.

Chapter 4: Connected Everywhere — WeChat, Telegram & More

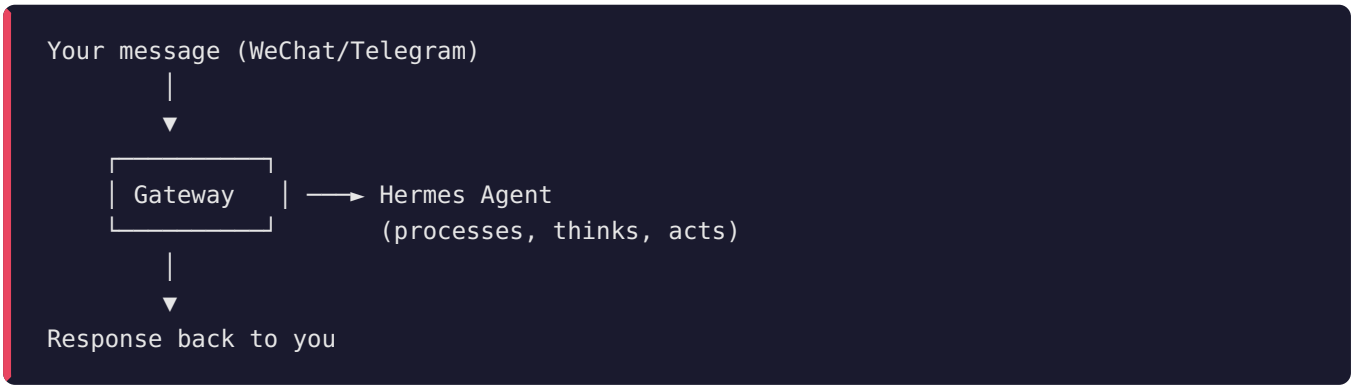
> **"Your AI agent in your pocket. Text it from anywhere, anytime."**

A CLI-only AI agent is useful. An AI agent you can text from your phone while grocery shopping? That's a game changer.

This chapter covers: - **WeChat** (Weixin) — works on your phone and MacBook - **Telegram** — the easiest option, works everywhere - **CLI tips** — when you're at your computer

4.1 The Gateway: How It Works

Hermes uses something called the **Gateway**. Think of it as a traffic controller:



The Gateway runs as a **background service** on your server. It listens for incoming messages from WeChat, Telegram, or other platforms, feeds them to Hermes, and sends the response back.

4.2 Setting Up WeChat (Weixin)

WeChat support in Hermes works through the official Weixin (微信) platform. Here's how to set it up:

Prerequisites

- A WeChat/Weixin account (you already have one)
- A WeChat Official Account (服务号) — required for bot functionality
- The Gateway service running on your server

Step 1: Create a WeChat Official Account

1. Go to mp.weixin.qq.com
2. Register a **Service Account (服务号)** — this is the type that supports API access
3. Complete the verification process (may take 1-3 business days)

Step 2: Configure in Hermes Gateway

Run the gateway setup wizard:

```
hermes gateway setup
```

Select **Weixin** and follow the prompts. You'll need: - Your AppID and AppSecret (from the WeChat Official Account dashboard) - A token (you create this yourself — anything, just remember it) - Your server's public IP or domain name

Step 3: Configure the Webhook URL

In your WeChat Official Account dashboard: 1. Go to **Settings** → **Development** → **Basic Configuration** 2. Set the **Server URL** to: `http://your-server-ip:9090/webhooks/weixin` 3. Set the **Token** to match what you configured in Hermes 4. Choose **AES-256** encryption mode 5. Submit (WeChat will send a verification request — Hermes handles it automatically)

Step 4: Start the Gateway

```
hermes gateway start
```

Check it's running:

```
hermes gateway status
```

Step 5: Scan to Chat

Go to your WeChat Official Account and send a message. If everything's set up correctly, you'll get a response from your Hermes Agent.

> ⚠️ **Common issue:** WeChat Official Accounts outside mainland China may have restrictions. If you can't get a Service Account, use the **Telegram** method below instead — it's simpler and works globally.

4.3 Setting Up Telegram (Simpler Alternative)

Telegram is the easiest messaging platform to set up with Hermes. It takes about 5 minutes.

Step 1: Create a Telegram Bot

1. Open Telegram and search for **@BotFather** (the official bot that creates bots)
2. Start a chat and send: `/newbot`
3. Choose a name (e.g., "My Hermes Agent")
4. Choose a username (must end in `bot`, e.g., `my_hermes_bot`)
5. BotFather will give you a **token** — save this!

Step 2: Configure in Hermes

Run:

```
hermes gateway setup
```

Select **Telegram** and paste your bot token when prompted.

Step 3: Start the Gateway

```
hermes gateway start
```

Step 4: Talk to Your Bot

Open Telegram, search for your bot's username, and send a message. Your Hermes Agent will respond.

That's it. No webhooks, no domain names, no verification delays. Telegram is truly plug-and-play.

4.4 The CLI: When You're at Your Computer

Sometimes you want to interact directly on the server. The CLI is great for:

- **Quick queries** without pulling out your phone: `bash hermes chat -q "What's the disk usage on this server?"`
- **Running background tasks**: `bash hermes chat -q "Research RAG techniques and save summary to ~/rag-notes.md"`
- **Administration** — configuring providers, managing skills, checking logs

Useful CLI Flags

```
hermes                      # Interactive chat session
hermes chat -q "query"      # One-shot question, non-interactive
hermes --continue           # Resume your last session
hermes -s skill-name        # Load a skill at startup
```

4.5 Pro Tip: Configure Per-Platform Settings

You can have different settings for different platforms. For example, use a cheap model for WeChat (quick answers) and a powerful model for CLI (complex tasks).

```
hermes skills config  # Shows platform-specific skill settings
```

This is covered in detail in Chapter 8 (Configuration Mastery).

4.6 Gateway Troubleshooting

Gateway won't start

Check the logs:

```
cat ~/.hermes/logs/gateway.log | tail -20
```

Most common causes: - Missing API keys in `.env` - Port already in use (change port in `config.yaml`) - Python dependencies not installed (run `hermes doctor`)

Gateway stops when I log out of SSH

You need to enable "lingering" for systemd user services:

```
sudo loginctl enable-linger $USER
```

This keeps systemd services running after you disconnect SSH.

Messages work but no response

Run `hermes doctor` to check for configuration issues:

```
hermes doctor
```

Your Hermes Agent is now on your phone. Try sending it a message from the supermarket: "Add milk and eggs to my shopping list." It remembers, because in Chapter 7, we cover the Skills System — the feature that makes Hermes learn and improve over time.

Chapter 5: Server Choice & Real-World Deployment

> **"Theory is great. But let me show you exactly what I run — two servers, two clouds, all working."**

I run Hermes on two different cloud providers. In this chapter, I'll walk you through both setups so you can choose what works best for you.

5.1 What Makes a Good Hermes Server

Hermes is lightweight. Here's the minimum:

Requirement	Minimum	Recommended
CPU	1 core	2 cores
RAM	512 MB	1-2 GB
Storage	5 GB	10-20 GB
OS	Ubuntu 20.04+	Ubuntu 22.04 LTS
Network	Public IP	Public IP + domain

Cost range: $4-12/month, depending on provider and region.

5.2 Option A: Tencent Cloud Lighthouse (My Setup)

I run Hermes on a Tencent Cloud Lighthouse instance. It's good value if you're in Asia.

Create the instance

1. Go to console.cloud.tencent.com/lighthouse
2. Click **Create Instance**
3. Choose: - **Region:** Closest to you (e.g., Singapore, Hong Kong) - **Image:** Ubuntu 22.04 LTS - **Plan:** The cheapest tier ($5-6/month) is sufficient
4. Set a **strong password** (save this!) or upload an SSH key
5. Click **Buy Now**

Configure SSH access

From your local machine:

```
ssh root@your-lighthouse-ip
```

Enter the password you set during creation.

Install Hermes

```
# 1. Update the system
apt update && apt upgrade -y

# 2. Install curl (if missing)
apt install curl -y

# 3. Install Hermes
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | ba

# 4. Reload shell
source ~/.bashrc
```

Start chatting

```
hermes
```

> **Note:** On Lighthouse, the default user is `root`. Hermes can run as root without issues, but for better security, consider creating a non-root user:
>
> ```bash adduser agentuser usermod -aG sudo agentuser
>
> # Then SSH as agentuser and install Hermes from that account
>
> ```

5.3 Option B: Google Cloud Platform (My Second Setup)

I also run Hermes on GCP. This is a good option if you need Google Cloud's network or want to use other GCP services.

Create the VM

1. Go to console.cloud.google.com
2. Navigate to **Compute Engine** → **VM Instances**
3. Click **Create Instance**
4. Configure: - **Name:** `hermes-agent` (or anything) - **Region/Zone:** Closest to you - **Machine type:** `e2-micro` (free tier eligible!) or `e2-small` (~$12/mo) - **Boot disk:** Ubuntu 22.04 LTS, 10 GB standard persistent disk - **Firewall:** Allow HTTP and HTTPS traffic

5. Click **Create**

SSH into the VM

From the GCP Console, click the **SSH** button next to your VM. Or from your local terminal:

```
gcloud compute ssh hermes-agent
```

Install Hermes

Same process as above:

```
sudo apt update && sudo apt upgrade -y
sudo apt install curl -y
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | ba
source ~/.bashrc
```

Cost-Saving Tip: Preemptible VMs

On GCP, you can use **preemptible (spot) VMs** which cost 60-80% less. The downside: they can be terminated with 30 seconds notice. This is fine if you don't need 100% uptime.

To protect against termination, set up a systemd service for Hermes (covered in Chapter 6) — it auto-restarts when the VM comes back.

5.4 Option C: DigitalOcean (Simplest)

DigitalOcean is the easiest VPS provider if you're outside Asia.

1. Go to digitalocean.com
2. Create a **Droplet**: - **OS:** Ubuntu 22.04 LTS - **Plan:** $6/mo (1 GB RAM, 1 CPU, 25 GB SSD) - **Region:** Closest to you
3. Add your SSH key
4. Create the Droplet
5. SSH in and follow the install steps above

5.5 Security Basics

Whichever provider you choose, do these as soon as your server is running:

1. Update everything

```
sudo apt update && sudo apt upgrade -y
```

2. Set up a firewall

```
sudo ufw allow OpenSSH
sudo ufw enable
```

If you're running the Gateway:

```
sudo ufw allow 9090  # Hermes Gateway default port
```

3. Disable root password login (SSH key only)

```
# On your local machine, generate a key if you don't have one:
ssh-keygen -t ed25519

# Copy it to the server:
ssh-copy-id root@your-server-ip

# On the server, edit SSH config:
sudo nano /etc/ssh/sshd_config

# Set these:
# PermitRootLogin prohibit-password
# PasswordAuthentication no

# Restart SSH:
sudo systemctl restart sshd
```

4. Store API keys safely

All API keys go in `~/.hermes/.env`. Set restrictive permissions:

```
chmod 600 ~/.hermes/.env
```

5.6 Provider Selection Guide

If you're in...	Recommend
Asia	Tencent Cloud Lighthouse ($5/mo)
Anywhere	GCP e2-micro (free tier)
Anywhere	DigitalOcean ($6/mo)
Europe	Hetzner Cloud ($4/mo — cheapest option)
US	Linode / Vultr ($5/mo)

5.7 Verification Checklist

After setting up your server:

- [ ] Can SSH into the server
- [ ] `hermes` command starts the interactive session
- [ ] API key is set in `~/.hermes/.env`
- [ ] Model responds to queries
- [ ] Firewall is active
- [ ] SSH key authentication works
- [ ] `chmod 600 ~/.hermes/.env` is set

Your Hermes Agent now lives on a server that runs 24/7. In Chapter 6, we set it up as a proper background service so it auto-starts on reboot and never goes down.

Chapter 7: Skills System — How Your Agent Gets Smarter Over Time

> **"The single feature that makes Hermes Agent more powerful than any chatbot: it learns."**

7.1 What Are Skills?

Skills are **reusable procedures** that Hermes saves and remembers. When you teach your agent how to do something once, it can do it again without being re-taught.

Think of skills like recipes in a cookbook: - A recipe tells you how to make a dish - A skill tells Hermes how to accomplish a task - Once saved, anyone (you or your agent) can use it

Real Example

Without skills:

> You: "Check if Amazon dropped the price on this laptop." Hermes: *scrapes Amazon, checks price, reports back*

Next time you ask:

> You: "Check if Amazon dropped the price on this laptop." Hermes: *has to figure it out all over again*

With skills:

> You: "Check if Amazon dropped the price on this laptop. Save this as a skill called 'price-watch'." Hermes: *does the work, saves the procedure*

Next time you ask:

> You: "Run price-watch on the new iPhone." Hermes: *already knows the steps. Instant.*

7.2 How Skills Work Under the Hood

A skill is just a markdown file with a special format:

```
~/.hermes/skills/
├── price-watch/
│   ├── SKILL.md          # The procedure description
```

```
│   └── references/         # Optional helper files
│       └── config.yaml
├── daily-briefing/
│   └── SKILL.md
└── research-agent/
    └── SKILL.md
```

Each `SKILL.md` contains: 1. **Metadata** — name, description, tags 2. **Instructions** — step-by-step what to do 3. **Edge cases** — what to watch out for

When you load a skill, Hermes reads this file and follows the instructions. It's like giving your agent a checklist + manual.

7.3 Loading a Skill

During a chat session:

```
/skill price-watch
```

Or at startup:

```
hermes -s price-watch,daily-briefing
```

7.4 Building Your First Skill

Let's create a skill that does something useful: a **daily briefing** that sends you weather + news every morning.

Create the skill directory and file:

```
mkdir -p ~/.hermes/skills/daily-briefing
nano ~/.hermes/skills/daily-briefing/SKILL.md
```

Paste this content:

```
---
name: daily-briefing
description: Sends a morning briefing with weather, news, and reminders
tags: [daily, briefing, weather, news]
---

# Daily Briefing Skill

## When to use
- Every morning when the user says "send the briefing"
- Scheduled via cron to run at 7 AM daily

## What to do

1. **Get the weather:**
   - Use `web_search` to find today's weather for the user's city
   - If city unknown, check memory or ask
```

```
2. **Get top news:**
   - Search for top 3 news headlines
   - Summarize each in 1 sentence

3. **Check calendar (if connected):**
   - List any events for today

4. **Format the response:**
   - Title: "☀️ Good Morning! Here's your briefing:"
   - Section 1: Weather (emoji + temp + conditions)
   - Section 2: News (3 bullet points)
   - Section 3: Calendar (events, if any)

5. **Deliver:**
   - Send back to the user's home channel
```

Now test it:

```
hermes -s daily-briefing
```

Then ask: *"Run the daily briefing."*

7.5 Installing Community Skills

Hermes has a catalog of community-created skills. Browse them:

```
hermes skills browse
```

Or search:

```
hermes skills search automation
```

Install one:

```
hermes skills install <skill-id>
```

Popular community skills to try:

Skill	What it does
`systematic-debugging`	Root-cause debugging framework
`writing-plans`	Creates implementation plans
`test-driven-development`	TDD workflow
`github-code-review`	PR code review automation

7.6 Auto-Learning: How Hermes Creates Skills on Its Own

Here's the powerful part: Hermes can save skills **automatically**. After completing a complex task (5+ tool calls), you can tell it:

> "Save that as a skill."

Or it can detect that a workflow is worth saving. When it does, it creates a `SKILL.md` with: - The problem it solved - The steps it took - The tools it used - Pitfalls to avoid

This means over time, your agent gets better and better at *your* specific tasks.

7.7 Enabling/Disabling Skills

Some skills should only run on certain platforms:

```
hermes skills config
```

This lets you: - Enable research skills only in CLI - Disable destructive skills on WeChat - Keep productivity skills everywhere

7.8 Skill Management Cheat Sheet

```
hermes skills list                # See all installed skills
hermes skills search <query>      # Search skill catalog
hermes skills install <id>        # Install from catalog
hermes skills update              # Update all outdated skills
hermes skills check               # Check for updates
hermes skills config              # Per-platform skill settings
/skill <name>                     # Load a skill mid-session
```

7.9 What to Teach Your Agent First

Here are 5 skills worth creating on day one:

1. **daily-briefing** — weather + news + calendar every morning
2. **web-researcher** — given a topic, produce a structured report
3. **file-organizer** — clean up downloads, sort by type/date
4. **server-health** — check disk, memory, uptime, send report
5. **note-taker** — save meeting notes to a formatted file

This is what separates Hermes from every other AI tool. Your agent doesn't just answer questions — it gets better over time. Every task you teach it makes the next one faster.

Chapter 8 covers Configuration Mastery — the full `config.yaml` breakdown, per-channel settings, model fallbacks, and credential pools.

Chapter 9: Automation Recipes — 10 Copy-Paste Ready Workflows

> **"The best part of running your own AI agent: automate anything. Here are 10 recipes you can deploy today."**

Each recipe in this chapter follows the same format: what it does, how to set it up, and the exact commands to run.

Recipe 1: Daily News Briefing → WeChat Every Morning

What it does: Sends a 3-item news summary + weather to your phone at 7 AM.

Step 1: Save the skill (from Chapter 7 — create `daily-briefing` skill first).

Step 2: Create a cron job:

```
hermes cron create "0 7 * * *" \
  --name "morning-briefing" \
  --prompt "Run the daily-briefing skill and send the result to WeChat" \
  --skills daily-briefing
```

Verify:

```
hermes cron list
```

> **Pro tip:** Run it once manually to test before the 7 AM trigger: `bash hermes cron run <job-id>`

Recipe 2: Website Change Detector

What it does: Checks a website daily and alerts you if content changed.

Create a skill called `site-watcher` that does:

1. Fetch the page content
2. Compare with a saved hash
3. If different → notify + update the hash

Set it up:

```
hermes cron create "0 9 * * *" \
  --name "site-watcher" \
  --prompt "Check https://example.com/pricing for changes. Fetch the page, compare with ~/site-hash
```

Recipe 3: Stock/Price Alert (Custom)

What it does: Checks a specific stock or product price on schedule.

```
hermes cron create "0 */4 * * *" \
  --name "price-watch-AAPL" \
  --prompt "Check the current price of AAPL stock. If it dropped more than 3% since yesterday's clo
```

For a product on Amazon, replace the prompt with:

```
"Check the price of [product URL]. If it's below $50, send me an alert with the price and link."
```

Recipe 4: Server Health Report

What it does: Emails or messages you a server health check every morning.

```
hermes cron create "0 6 * * *" \
  --name "server-health" \
  --prompt "Check the server health: run 'df -h' for disk, 'free -h' for memory, 'uptime' for syste
```

Recipe 5: RSS Feed → AI Summary → Chat

What it does: Monitors an RSS/Atom feed and sends you AI-summarized updates.

```
hermes cron create "0 8,18 * * *" \
  --name "rss-watcher" \
  --prompt "Fetch and parse this RSS feed: https://hnrss.org/frontpage. For the top 5 stories, writ
```

Recipe 6: Email Auto-Summarizer

What it does: Summarizes your unread emails and flags urgent ones.

> **Requires:** Email integration via Hermes Gateway

```
hermes cron create "0 9,14 * * 1-5" \
  --name "email-summary" \
  --prompt "Check my email inbox. Find the 5 most important unread emails. Summarize each: sender,
```

Recipe 7: GitHub Repository Watcher

What it does: Checks your starred repos for new releases/issues.

```
hermes cron create "0 10 * * 1" \
  --name "github-watch" \
  --prompt "Check the GitHub repo NousResearch/hermes-agent for: (1) new releases in the last week,
```

Recipe 8: Language Learning Flashcards

What it does: Sends you daily vocabulary in your target language.

```
hermes cron create "0 20 * * *" \
  --name "japanese-vocab" \
  --prompt "Generate 5 Japanese vocabulary words for a beginner. Format each as: word (in Japanese)
```

Change "Japanese" to your target language and "beginner" to your level.

Recipe 9: Weekly Research Report

What it does: Researches a topic and saves a report every Sunday.

```
hermes cron create "0 9 * * 0" \
  --name "weekly-research" \
  --prompt "Research the latest developments in [your field/interest]. Search for 5 recent articles
```

Recipe 10: Custom Research Agent

What it does: Takes a topic, produces a structured markdown report.

This works as a skill rather than a cron job. Create `~/.hermes/skills/deep-research/SKILL.md`:

```
---
name: deep-research
description: Given a topic, produces a structured research report
tags: [research, report, analysis]
---

# Deep Research Skill

## How to use
Say: "Research [topic] and write a report"

## Steps

1. Search the web for the topic (3-5 different searches from different angles)
2. Read and extract key information from each source
3. Organize into sections:
   - Overview (2-3 paragraphs)
   - Key Findings (bullet points)
   - Notable Details
   - Contrarian Views or Debates
   - Sources (with links)
4. Write to `~/research/<topic-slug>-<date>.md`
5. Summarize the report to the user
```

Usage: `/skill deep-research` then *"Research serverless AI inference options in 2025."*

Cron Job Management Cheat Sheet

```
hermes cron list                            # List all jobs
hermes cron run <job-id>                    # Run job immediately
hermes cron pause <job-id>                  # Pause a job
hermes cron resume <job-id>                 # Resume a paused job
hermes cron remove <job-id>                 # Delete a job

# Schedule formats:
hermes cron create "0 7 * * *" ...          # Every day at 7 AM
hermes cron create "*/30 * * * *" ...       # Every 30 minutes
hermes cron create "0 9 * * 1" ...          # Every Monday at 9 AM
hermes cron create "2025-06-01T09:00:00" ...  # One-time on June 1
hermes cron create "30m" ...                # Every 30 minutes (simple)
hermes cron create "every 2h" ...           # Every 2 hours (simple)
```

---

10 recipes. Zero ongoing effort. Set them up once, and your agent does the work forever.

Chapter 11 covers troubleshooting — the real-world problems you'll encounter and exactly how to fix them.

Chapter 11: Troubleshooting Common Problems

> **"Everything that can go wrong — and exactly how to fix it."**

This chapter collects real problems I encountered while running Hermes across two different servers. If something breaks, check here first.

11.1 Installation Problems

"curl: command not found"

```
# Ubuntu/Debian
sudo apt update && sudo apt install curl -y

# CentOS/RHEL
sudo yum install curl -y
```

"Permission denied" during install

Run as a user with sudo access, or:

```
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | su
```

"hermes: command not found" after install

The installer updates `~/.bashrc` but your current session doesn't see it:

```
source ~/.bashrc
# OR
exec $SHELL -l
```

If it still doesn't work, check if the install actually ran:

```
ls -la ~/.hermes/
```

If the directory is empty or missing, re-run the install.

Install hangs or is very slow

Some VPS providers have slow connections to GitHub. Try:

```
# First, update your system
sudo apt update && sudo apt install python3 python3-venv git -y

# Then manually clone and install
git clone https://github.com/NousResearch/hermes-agent.git ~/.hermes/hermes-agent
cd ~/.hermes/hermes-agent
python3 -m venv venv
source venv/bin/activate
pip install -e .
```

11.2 Model / Provider Issues

"No models provided" or HTTP 400

This usually means the model name doesn't match what the provider expects.

Fix: Check the exact model name:

```
hermes config set model.default gemini-2.5-flash
```

Common correct names: - Google: `gemini-2.5-flash`, `gemini-2.5-pro` - OpenRouter: `openai/gpt-4o-mini`, `anthropic/claude-sonnet-4` - OpenAI: `gpt-4o`, `gpt-4o-mini` - Anthropic: `claude-sonnet-4`, `claude-haiku-4`

"401 Unauthorized" or "Invalid API key"

Your `.env` file has a wrong or missing key.

```
# Check if the key exists
cat ~/.hermes/.env | grep API_KEY

# If it's there, check for typos
# If it's missing, add it
echo 'GOOGLE_API_KEY=*** >> ~/.hermes/.env

# Verify the config can read it
hermes doctor
```

"Rate limit exceeded"

You hit the provider's rate limit. Solutions:

1. **Wait a minute** and try again
2. **Switch to a different model** temporarily
3. **Set up fallback models** (see Chapter 3.7)

Provider works in web UI but not in Hermes

Some providers (like DeepSeek reasoning models) return extra fields (like `reasoning_content`) that Hermes doesn't handle. **Use the standard chat model version instead:**

```
# ❌ Don't use "deepseek-reasoner" (has thinking mode issues)
# ✅ Use "deepseek-chat" (standard chat, no problems)
hermes config set model.default deepseek-chat
```

11.3 Gateway Problems

Gateway won't start

```
# Check logs
cat ~/.hermes/logs/gateway.log | tail -20

# Common fix: restart
hermes gateway restart

# If still failing, reset
systemctl --user reset-failed hermes-gateway
hermes gateway restart
```

Gateway dies when I log out of SSH

Systemd user services don't survive logout by default:

```
sudo loginctl enable-linger $USER
```

This keeps the gateway running after you disconnect.

WeChat not responding

Check:

```
# 1. Is the gateway running?
hermes gateway status

# 2. Is the platform configured?
hermes gateway setup

# 3. Is the webhook URL correct?
# Check in mp.weixin.qq.com → Settings → Basic Configuration
# URL should be: http://your-server-ip:9090/webhooks/weixin
```

Telegram bot not responding

```
# 1. Is the bot token correct?
# Go to @BotFather on Telegram and /mybots → API Token

# 2. Is the gateway running?
hermes gateway status
```

```
# 3. Check gateway logs for errors
cat ~/.hermes/logs/gateway.log | grep -i error | tail -10
```

11.4 Cron Job Issues

Cron job didn't run

```
# List jobs to confirm it's still there
hermes cron list

# Check if the scheduler is running
hermes gateway status

# Run it manually to test
hermes cron run <job-id>
```

Cron job ran but no output received

The job ran but couldn't deliver. Check:

1. Is the Gateway running? (delivery needs the gateway)
2. Is the home channel set correctly?
3. Try: `hermes cron run <job-id> --deliver weixin`

Cron job "Model not responding"

Your chosen provider was rate-limited or had an outage. Solutions: - Add fallback models (Chapter 3.7) - Use a different provider for cron jobs - Schedule retries

11.5 General Errors

Message too long

Hermes returns a very long response and it gets truncated on WeChat/Telegram.

Fix: Add to your prompt:

> "Be concise. Keep responses under 200 words."

Or set it in skills:

```
# In your skill's instructions
- Keep all responses under 150 words
- Use bullet points, not paragraphs
```

"Session not found"

```
hermes sessions list          # Find the session
hermes sessions browse        # Interactive picker
hermes --continue             # Resume most recent
```

"Tool not available"

Some tools need to be enabled per-platform:

```
hermes tools                  # Interactive toggle
hermes tools enable web       # Enable web tool
hermes tools enable terminal  # Enable terminal tool
```

After changes, start a new session: `/reset`

Memory not working

```
# Check memory status
hermes memory status

# Enable it
hermes config set memory.memory_enabled true

# Enable user profile
hermes config set memory.user_profile_enabled true
```

11.6 The Ultimate Reset

If nothing works, reset Hermes without losing your skills:

```
# 1. Save your API keys
cp ~/.hermes/.env ~/env-backup.txt

# 2. Back up skills
cp -r ~/.hermes/skills ~/skills-backup

# 3. Reinstall
hermes uninstall
curl -fsSL https://raw.githubusercontent.com/NousResearch/hermes-agent/main/scripts/install.sh | ba

# 4. Restore
cp ~/env-backup.txt ~/.hermes/.env
cp -r ~/skills-backup/* ~/.hermes/skills/

# 5. Reconfigure
hermes setup
```

Quick Reference: Error to Fix

Error Message	Fix
`command not found: hermes`	`source ~/.bashrc`
`401 Unauthorized`	Check API key in `.env`
`HTTP 400`	Wrong model name
`Rate limit exceeded`	Wait or switch models
`Gateway not running`	`hermes gateway start`
`Cron job didn't deliver`	Check gateway status
`No models provided`	Check `model.default` in config
`session_search failed`	Check memory provider config
`Tool not available`	`hermes tools enable [name]`

That covers every major problem I've encountered. If you hit something not listed here, run `hermes doctor` and check the logs — both will point you in the right direction.