KI-Agenten auf dem eigenen PC installieren

Claude Code, Hermes Agent und OpenClaw unter Windows 11 – zweimal: zuerst direkt in Windows (PowerShell), danach in Linux (WSL mit Ubuntu). Mit allen Befehlen zum Abtippen. Das Video im Workshop zeigt genau diese Schritte und endet jeweils vor der Anmeldung.

Überblick

Werkzeug	Was es ist	Wann es läuft
Claude Code (Anthropic)	Ein KI-Assistent im Terminal-Fenster. Er liest und bearbeitet Dateien auf Ihrem PC und führt Aufgaben aus, die Sie ihm in normaler Sprache geben.	Solange das Fenster offen ist.
Hermes Agent (Nous Research)	Ein Open-Source-Agent, der mit KI-Modellen verschiedener Anbieter arbeitet. Über ein „Gateway“ ist er auch per Messenger (z. B. Telegram) erreichbar.	Im Fenster: solange es offen ist. Als Gateway-Dienst: solange der Rechner läuft.
OpenClaw	Ein Open-Source-Agent, der um ein dauerhaft laufendes Gateway gebaut ist. Sie schreiben ihm z. B. über Telegram oder WhatsApp, er antwortet und erledigt Aufgaben.	Gedacht für den Betrieb rund um die Uhr – dafür muss der Rechner durchgehend laufen.
WSL mit Ubuntu (Microsoft)	Das „Windows-Subsystem für Linux“: ein vollwertiges Linux innerhalb von Windows. Selbst kein Agent, sondern die Grundlage für Teil 3.	Startet bei Bedarf, wenn Sie ein Ubuntu-Fenster öffnen.

Direkt in Windows oder in WSL? Alle drei Werkzeuge gibt es in beiden Varianten. Das Handout zeigt beide Wege; Sie brauchen für den Anfang nur einen.

Direkt in Windows (Teil 1) ist der schnellere Einstieg: keine Administratorrechte, kein Neustart, keine Virtualisierung. Die Agenten arbeiten direkt mit Ihren Windows-Ordnern und -Programmen. Gut zum Kennenlernen.

In WSL (Teil 2 und 3) lohnt sich für den Dauerbetrieb: Die OpenClaw-Doku nennt WSL die „Linux-kompatibelste“ Laufzeit unter Windows, und es gelten dieselben Befehle wie auf Mac und Linux. Dafür braucht es einmal Administratorrechte und einen Neustart.

Beides zugleich geht. Es sind getrennte Installationen mit getrennter Anmeldung und getrennten Einstellungen. Verbinden Sie einen Telegram-Bot oder eine WhatsApp-Nummer aber nur mit **einer** Installation.

Was Sie brauchen

- **Windows 11** oder Windows 10 (64-Bit), Internet. Git, Node.js oder Python müssen Sie nicht vorher installieren – die Installer holen sich, was fehlt.
- **Administratorrechte** nur für die WSL-Installation (Teil 2). Die Installationen in Teil 1 brauchen keine. Am Firmen-PC ggf. die IT fragen.
- Nur für WSL: **Virtualisierung** muss eingeschaltet sein (bei den meisten PCs ab Werk; sonst im BIOS/UEFI).
- **Ein Konto für die Nutzung** – die Installation selbst ist kostenlos:

Werkzeug	Was Sie für die Nutzung brauchen (laut Herstellerdoku)
Claude Code	Ein Abo **Claude Pro, Max, Team oder Enterprise** oder ein **Claude-Console-Konto** (API, Vorauszahlung, nach Verbrauch). **Der kostenlose Claude-Plan reicht nicht.** Beim Abo teilen sich Claude und Claude Code dieselben Nutzungsgrenzen.
Hermes Agent	Ein Modell-Anbieter Ihrer Wahl, z. B. Nous Portal (eigenes Abo), OpenRouter, ein API-Schlüssel. **Achtung Claude-Abo:** Mit Claude Pro geht die Anmeldung nicht; mit Claude Max nur über zusätzlich gekaufte Extra-Credits – Sie zahlen also extra.
OpenClaw	Entweder ein vorhandener Claude-Code-Login (verbraucht Ihr Abo-Kontingent) oder ein API-Schlüssel (Kosten je Nutzung). Für Telegram ein Bot-Token, für WhatsApp ein Smartphone.

Aktuelle Preise: claude.com/pricing bzw. die Seite des jeweiligen Anbieters. Ein Agent, der rund um die Uhr läuft, kann auch rund um die Uhr Kosten verursachen.

Zwei Fenster, zwei Welten. PowerShell (Windows) erkennen Sie am **PS** am Zeilenanfang: `PS C:\Users\Name>`. Das Ubuntu-Fenster (Linux) zeigt `name@rechner:~$`. Jeder Kasten sagt, wo der Befehl hingehört. Befehle vollständig abtippen oder kopieren und mit Enter bestätigen.

Teil 1 – Direkt in Windows (PowerShell)

Für alle drei Werkzeuge: Startmenü → „PowerShell“ eingeben → **Windows PowerShell** öffnen (nicht „Windows PowerShell (x86)“). Administratorrechte sind nicht nötig.

1.1 Claude Code

1 Installer starten. Er zeigt keinen Fortschritt – bitte warten, nicht abbrechen und nicht doppelt starten. Fertig ist er bei **„Claude Code successfully installed!“**.

PowerShell
```
irm https://claude.ai/install.ps1 | iex
```

2 PowerShell schließen, **ein neues Fenster öffnen** und prüfen. Es erscheint eine Versionsnummer mit „(Claude Code)“.

PowerShell – neues Fenster
```
claude --version
```

ANMELDUNG Erst jetzt: `claude` eintippen. Es öffnet sich ein Browser, dort melden Sie sich mit Ihrem Claude-Konto an (Abo oder Console). Öffnet sich kein Browser, drücken Sie **c** – das kopiert den Anmeldelink.

1.2 Hermes Agent

1 Installer starten – in dieser Form **ohne** den automatischen Einrichtungsassistenten (so wie im Video). Der Befehl ist **eine** Zeile. Das dauert mehrere Minuten, weil Hermes Git, Python, Node.js und einen Browser selbst herunterlädt. Fertig ist er bei **„Hermes Agent install complete“**.

PowerShell – eine Zeile
```
& ([scriptblock]::Create((irm https://hermes-agent.nousresearch.com/install.ps1))) -NonInteractive
```

2 Ein **neues PowerShell-Fenster** öffnen und prüfen. `Get-Command hermes` zeigt einen Pfad unter …\AppData\Local\hermes\bin, dort liegt auch alles Weitere von Hermes.

PowerShell – neues Fenster
```
Get-Command hermes
hermes --version
```

ANMELDUNG `hermes model` – Anbieter und Modell wählen, API-Schlüssel eingeben oder im Browser anmelden. Alternativ der ganze Assistent: `hermes setup` (mit Nous-Portal-Konto: `hermes setup --portal`). Telegram & Co. danach mit `hermes setup gateway`. Dafür sind keine Administratorrechte nötig.

1.3 OpenClaw

1 Installer starten, wieder **ohne** automatische Einrichtung und als **eine** Zeile. Fehlen Node.js oder Git, holt der Installer sie selbst. Bittet Windows dabei um eine Bestätigung, bestätigen Sie sie.

PowerShell – eine Zeile
```
& ([scriptblock]::Create((iwr -useb https://openclaw.ai/install.ps1))) -NoOnboard
```

2 Ein **neues PowerShell-Fenster** öffnen und prüfen. Meldet PowerShell, die Ausführung von Skripts sei deaktiviert, tippen Sie stattdessen `openclaw.cmd --version` (siehe Fehlerliste).

PowerShell – neues Fenster
```
openclaw --version
```

ANMELDUNG `openclaw onboard --install-daemon` startet die Einrichtung: KI-Zugang wählen (vorhandener Claude-Code-Login oder API-Schlüssel), dann Kanäle verbinden – Telegram mit einem Bot-Token von @BotFather, WhatsApp per QR-Code. Unter Windows wird der Dienst als geplante Aufgabe angelegt. Soll er schon beim Systemstart laufen, diesen Befehl in einer PowerShell **als Administrator** ausführen; sonst startet er erst, wenn Sie sich anmelden.

Teil 2 – WSL mit Ubuntu einrichten

1 Startmenü → „PowerShell“ → **Rechtsklick → „Als Administrator ausführen“**.

PowerShell als Administrator
```
wsl --install
```

2 Den Rechner **neu starten** (über das Startmenü oder mit diesem Befehl):

PowerShell
```
Restart-Computer
```

3 Nach dem Neustart: wieder eine PowerShell öffnen – diesmal genügt eine **normale**. Erst jetzt lädt Windows **Ubuntu** herunter; vorher steht es noch nicht im Startmenü.

PowerShell – nach dem Neustart
```
wsl --install -d Ubuntu
```

Ubuntu fragt nach einem **Benutzernamen** und schlägt Ihren Windows-Namen vor. Passt er (Kleinbuchstaben, ohne Leerzeichen und Umlaute), mit Enter bestätigen. Dann zweimal ein **Passwort**. Beim Tippen erscheint nichts – das ist normal. Das ist ein lokales Linux-Konto, kein Online-Konto. **Passwort merken**, es wird später für sudo gebraucht. Zum Schluss fragt Ubuntu, ob es anonyme Systemberichte senden darf („platform metrics“). Das ist freiwillig: das vorgegebene y löschen und n eingeben.

4 Ubuntu aktualisieren und die Werkzeuge curl, git und libatomic1 (braucht Hermes) sicherstellen. Fragt Ubuntu nach dem Passwort, Ihr Linux-Passwort eingeben (diesmal erscheint je Zeichen ein Sternchen).

Ubuntu (WSL)
```
sudo apt update && sudo apt upgrade -y
sudo apt install -y curl git libatomic1
```

5 Kontrolle: In der Liste steht Ubuntu mit **VERSION 2**.

PowerShell
```
wsl -l -v
```

Teil 3 – In WSL (Ubuntu)

Alle Befehle in diesem Teil gehören ins **Ubuntu-Fenster**, nicht in PowerShell: Startmenü → „Ubuntu“ öffnen und **einmal in das Fenster klicken**, damit es Ihre Eingaben annimmt. Was Sie in Teil 1 installiert haben, bleibt davon unberührt.

3.1 Claude Code

1 Den Linux-Installer starten:

Ubuntu (WSL)
```
curl -fsSL https://claude.ai/install.sh | bash
```

2 Nur falls der Installer meldet „~/.local/bin is not in your PATH“:

Ubuntu (WSL) – nur bei PATH-Hinweis
```
echo 'export PATH="$HOME/.local/bin:$PATH"' >> ~/.bashrc && source ~/.bashrc
```

3 Prüfen. `which claude` muss **/home/<name>/.local/bin/claude** zeigen, nicht /mnt/c/… (das wäre die Windows-Installation aus Teil 1).

Ubuntu (WSL)
```
claude --version
which claude
```

ANMELDUNG `claude` eintippen. Unter WSL zeigt der Browser nach der Anmeldung oft einen **Code**. Diesen kopieren und im Ubuntu-Fenster bei „Paste code here if prompted“ einfügen (Rechtsklick oder Umschalt+Einfg). Danach erscheint „Login successful“.

3.2 Hermes Agent

1 Installation ohne den automatischen Einrichtungsassistenten. Das dauert einige Minuten. Fehlt libatomic1 aus Teil 2, kann das mitgelieferte Node.js nicht starten.

Ubuntu (WSL)
```
curl -fsSL https://hermes-agent.nousresearch.com/install.sh | bash -s -- --non-interactive
```

2 Fenster neu laden und prüfen. Hermes legt hier alles unter ~/.hermes ab, getrennt von der Windows-Installation. `which hermes` muss einen Pfad unter /home/<name>/ zeigen, nicht /mnt/c/….

Ubuntu (WSL)
```
source ~/.bashrc
hermes --version
which hermes
```

ANMELDUNG Wie unter Windows: `hermes model` für Anbieter, Modell und Schlüssel, oder der ganze Assistent `hermes setup`. Telegram & Co. mit `hermes setup gateway`.

3.3 OpenClaw

1 Installation ohne automatische Einrichtung. Der Installer holt Node.js und weitere Werkzeuge selbst und fragt dafür ggf. nach Ihrem **Linux-Passwort**.

Ubuntu (WSL)
```
curl -fsSL https://openclaw.ai/install.sh | bash -s -- --no-onboard
```

2 Fenster neu laden und prüfen. Der Installer hat einen neuen Ordner eingetragen, den das offene Fenster erst danach kennt. `which openclaw` muss einen Pfad unter /home/<name>/ oder /usr/ zeigen, nicht /mnt/c/… (das wäre die Windows-Installation aus Teil 1).

Ubuntu (WSL)
```
source ~/.bashrc
openclaw --version
which openclaw
```

3 Optional, für den Dauerbetrieb als Hintergrunddienst: **systemd** in WSL. Zuerst nachsehen – steht dort schon „systemd=true“, ist nichts zu tun:

Ubuntu (WSL)
```
cat /etc/wsl.conf
```

4 Nur wenn die Zeile fehlt: ergänzen (die vier Zeilen nacheinander eingeben), danach WSL in PowerShell neu starten und Ubuntu wieder öffnen.

Ubuntu (WSL) – vier Zeilen
```
sudo tee -a /etc/wsl.conf >/dev/null <<'EOF'
[boot]
systemd=true
EOF
```

PowerShell
```
wsl --shutdown
```

ANMELDUNG `openclaw onboard --install-daemon` startet die Einrichtung: KI-Zugang wählen (vorhandener Claude-Code-Login oder API-Schlüssel), dann Kanäle verbinden – Telegram mit einem Bot-Token von @BotFather, WhatsApp per QR-Code als verknüpftes Gerät.

Wenn etwas nicht klappt

Meldung / Problem	Bedeutung und Lösung
Windows (PowerShell)	
„Die Benennung ‚claude‘ (bzw. ‚hermes‘, ‚openclaw‘) wurde nicht … erkannt“	Zuerst ein **neues** PowerShell-Fenster öffnen – der neue Pfad gilt erst dort. Hilft das bei Claude Code nicht: PATH-Reparatur unten.
„Der Befehl ‚irm‘ ist entweder falsch geschrieben oder konnte nicht gefunden werden“	Sie sind in der Eingabeaufforderung (CMD), nicht in PowerShell. PowerShell öffnen.
„Es wurde kein Parameter gefunden, der dem Parameternamen ‚NonInteractive‘ (bzw. ‚NoOnboard‘) entspricht“	Der Schalter hängt an der Kurzform `iex (irm …)`. Dort geht das nicht: die lange Form mit `& ([scriptblock]::Create(…))` aus Teil 1 verwenden.
„Es wurde kein Parameter gefunden … ‚fsSL‘“ oder ‚bash‘ wird nicht erkannt	Ein Linux-Befehl aus Teil 3 wurde in PowerShell eingegeben. Er gehört ins **Ubuntu-Fenster**.
„… kann nicht geladen werden, da die Ausführung von Skripts auf diesem System deaktiviert ist“	Betrifft `openclaw` bzw. `npm`. Entweder `openclaw.cmd` statt `openclaw` tippen, oder einmalig (ohne Administrator): `Set-ExecutionPolicy -Scope CurrentUser RemoteSigned`
„The assignment expression is not valid“ (Hermes)	Die heruntergeladene Datei beginnt mit einem unsichtbaren Zeichen (BOM). Dann die Kurzform `iex (irm https://hermes-agent.nousresearch.com/install.ps1)` verwenden; sie startet am Ende den Assistenten.
„Node.js installation may require a terminal restart“ (OpenClaw)	Neues PowerShell-Fenster öffnen und den Installer noch einmal starten.
„Claude Code does not support 32-bit Windows“	Sie haben „Windows PowerShell (x86)“ erwischt. Die normale PowerShell öffnen.
Installer zeigt lange nichts Neues	Normal. Warten, nicht ein zweites Mal starten. Hermes schreibt Einzelheiten nach %LOCALAPPDATA%\hermes\logs\install.log.
„… being used by another process“ (Claude Code)	Doppelstart oder Virenscanner. Andere Installer-Fenster schließen, den Ordner %USERPROFILE%\.claude\downloads löschen, neu starten.

Meldung / Problem	Bedeutung und Lösung
Hermes wieder entfernen	`hermes uninstall` verwenden, nicht einfach den Ordner löschen.
WSL (Ubuntu)	
WSL: Fehler 0x80370102	Virtualisierung fehlt oder ist aus. Am eigenen PC im BIOS/UEFI einschalten. In einer Cloud-VM braucht es „Nested Virtualization“.
`wsl --install` zeigt nur einen Hilfetext	WSL ist schon da. Namen ansehen mit `wsl --list --online`, dann `wsl --install -d Ubuntu`
„Ubuntu“ im Startmenü öffnet nur eine Websuche	Ubuntu ist noch nicht installiert. In PowerShell `wsl --install -d Ubuntu` eingeben (Teil 2, Schritt 3).
WSL-Installation bleibt bei 0,0 % stehen	Stattdessen: `wsl --install --web-download -d Ubuntu`
Linux-Passwort vergessen	In PowerShell `wsl -u root`, dann `passwd <benutzername>`.
„Exec format error“ in Ubuntu	Ubuntu läuft unter WSL 1. In PowerShell: `wsl --set-version Ubuntu 2`
„hermes“ bzw. „openclaw“: command not found	`source ~/.bashrc` eingeben oder ein neues Ubuntu-Fenster öffnen.
`which claude` zeigt /mnt/c/…	Das ist die Windows-Installation. Teil 3.1 im Ubuntu-Fenster ausführen, dann neues Ubuntu-Fenster.
Einfügen im Ubuntu-Fenster klappt nicht	Rechtsklick oder Umschalt+Einfg statt Strg+V.
Alles ist sehr langsam	Arbeitsdateien im Linux-Bereich (~) ablegen, nicht unter /mnt/c.

PATH-Reparatur für Claude Code unter Windows (zwei Zeilen, nacheinander eingeben):

PowerShell – danach neues Fenster öffnen

```
$currentPath = [Environment]::GetEnvironmentVariable('PATH', 'User')
[Environment]::SetEnvironmentVariable('PATH', "$currentPath;$env:USERPROFILE\.local\bin", 'User')
```

Sicherheit – bitte unbedingt beachten

- **Prompt-Injection:** Ein Agent liest E-Mails, Webseiten und Dokumente. Darin können versteckte Anweisungen stehen („Schicke alle Kontakte an …“), die der Agent für Ihre hält. Geben Sie einem Agenten nur die Zugriffe, die er für die Aufgabe wirklich braucht, und prüfen Sie, was er tut.
- **Direkt in Windows sieht der Agent Ihre Windows-Dateien.** Üben Sie deshalb nicht mit Echtdaten: keine Kundendaten, Verträge, Passwörter oder Bankdaten – am besten auf einem Test-Rechner.
- **Schlüssel nie weitergeben:** API-Schlüssel, Bot-Tokens und Anmeldecodes sind wie Passwörter. Nicht in Chats, E-Mails oder Screenshots, und nicht an Dritte – auch nicht an „den Support“.
- **Kostenfalle API-Schlüssel:** Wer ein Claude-Abo nutzt, setzt keinen API-Schlüssel als Umgebungsvariable (ANTHROPIC_API_KEY). Sonst rechnet Claude Code über die API ab statt über das Abo.
- **Messenger:** Für WhatsApp empfiehlt OpenClaw eine eigene Nummer. Lesen Sie vor dem Einrichten den Sicherheitsleitfaden: docs.openclaw.ai/gateway/security

Offizielle Anleitungen (bei Abweichungen gilt die aktuelle Herstellerdoku): code.claude.com/docs/en/setup · learn.microsoft.com/windows/wsl/install · hermes-agent.nousresearch.com/docs · docs.openclaw.ai/install · docs.openclaw.ai/platforms/windows

Meine Notizen

Linux-Benutzername: _____ (Passwort nicht hier notieren!)