Nemeton

Benutzerhandbuch

Version 1.0 Beta — Oktober 2026

Jürgen Koller Software GmbH

Inhalt

Nemeton

Funktionen

Erste Schritte

Installation · Systemanforderungen · Erste VM erstellen

VM erstellen

Linux-VMs · macOS-VMs · Standardwerte

VM-Konfiguration

CPU und Arbeitsspeicher · Festplatte · Netzwerk · Display · Audio · Shared Folders (VirtioFS) · Rosetta 2 · USB-Geräte

VM-Verwaltung

Starten und Stoppen · Inline-Display und Fullscreen · Auto-Start · Zwischenablage · Notizen · Diagnose

macOS Auto-Setup

Setup-Assistent überspringen · Benutzerkonto · Auto-Login · Erststart-Aktionen · Konfigurationsoptionen

Snapshots

Copy-on-Write auf APFS · Snapshot erstellen · Snapshot wiederherstellen · Snapshot löschen · Einschränkungen

API & Automatisierung

Verbindung · REST-API · Echtzeit-Events · Webhooks · MCP (Model Context Protocol)

KI-Assistent

Überblick · Anbieter · Einwilligung und Einrichtung · Bestätigungsmodus

Claude Code in der VM

Überblick · Voraussetzungen · Starten und bedienen · VM steuert VM · Aufgaben und Protokolle

Tastenkürzel

Häufige Fragen

Kann ich macOS-VMs auf einem Intel-Mac erstellen? · Wie viele VMs kann ich gleichzeitig laufen lassen? · Warum belegt mein Disk-Image weniger Platz als konfiguriert? · Können Snapshots bei

laufender VM erstellt werden? · Wo werden die VM-Daten gespeichert? · Was kostet Nemeton?

Nemeton

Benutzerhandbuch — Version 1.0 Beta

Nemeton ist eine native macOS-App zur Verwaltung virtueller Maschinen auf Basis von Apples Virtualization.framework. Erstellen, konfigurieren und betreiben Sie VMs direkt auf Ihrem Mac — ohne zusätzliche Hypervisor-Software. Alle VMs laufen nativ auf Apple Silicon.

Funktionen

- **Linux- & macOS-VMs** — virtuelle Maschinen erstellen, konfigurieren und betreiben
- **9 Linux-Distributionen** — mit vorinstallierten Disk-Images und ISO-Downloads
- **Snapshots** — sofortige Copy-on-Write-Klone auf APFS
- **macOS Auto-Setup** — Setup-Assistent überspringen, Benutzerkonten vorkonfigurieren
- **REST-API & MCP** — REST-API mit OpenAPI-Dokumentation und über 200 MCP-Werkzeuge für Automatisierung und KI-Agenten
- **KI-Assistent** — mit Nemeton Cloud, eigenem API-Schlüssel oder Ollama lokal
- **Claude Code in der VM** — Sitzungen in macOS-VMs starten, mitlesen und bedienen (Beta)
- **Shared Folders** — Host-Ordner per VirtioFS einbinden

Erste Schritte

Installation

Nemeton ist als Direktdownload verfügbar. Die heruntergeladene DMG-Datei enthält die App, die per Drag & Drop in den Programme-Ordner gezogen wird.

Systemanforderungen

- **macOS 15 (Sequoia)** oder neuer
- **Apple Silicon** empfohlen (für macOS-VMs zwingend erforderlich)
- Ausreichend Festplattenspeicher für VM-Disk-Images (ab ca. 64 GB pro VM)

Erste VM erstellen

Nach dem Start zeigt Nemeton eine leere Seitenleiste. Über **Ablage** → **Neue Linux-VM** (⌘N) oder **Ablage** → **Neue macOS-VM** (⌘⇧N) wird eine neue virtuelle Maschine angelegt.

Für Linux-VMs wählen Sie eine Distribution aus der Liste. Nemeton bietet vorinstallierte Disk-Images für schnellen Start sowie ISO-Downloads für manuelle Installation. macOS-VMs laden automatisch das passende IPSW-Installationsimage von Apple.

Die VM erscheint in der Seitenleiste und kann sofort konfiguriert oder gestartet werden.

VM erstellen

Linux-VMs

Nemeton unterstützt die folgenden Linux-Distributionen:

Distribution	Version	Architektur
Fedora Workstation	43	ARM64
Ubuntu Server	24.04.4 LTS	ARM64
Ubuntu Desktop	24.04.4 LTS	ARM64
Debian	13.4.0	ARM64
Fedora Server	43	ARM64
Alpine Linux	3.21	ARM64
openSUSE Tumbleweed	Rolling	ARM64
Rocky Linux	9.5	ARM64
Arch Linux	Rolling	x86_64

Für Fedora Workstation steht ein vorinstalliertes Disk-Image bereit, das den ISO-Installationsprozess überspringt. Die VM bootet direkt in ein fertig eingerichtetes System.

Downloads verwenden eine automatische Retry-Logik mit drei Versuchen und exponentiellem Backoff bei Verbindungsproblemen.

macOS-VMs

macOS-VMs sind ausschließlich auf Apple-Silicon-Macs verfügbar. Nemeton lädt die passende IPSW-Datei automatisch von Apple herunter. Die Version wird anhand des Host-Systems ermittelt — es kann nur eine macOS-Version installiert werden, die der Host-Version entspricht oder älter ist.

Standardwerte

Einstellung	Linux	macOS
CPU-Kerne	4	4
Arbeitsspeicher	8 GB	8 GB
Festplatte	64 GB	80 GB
Netzwerk	NAT	NAT

VM-Konfiguration

CPU und Arbeitsspeicher

- **CPU-Kerne**: 1 bis zur Anzahl der physischen CPU-Kerne des Macs
- **Arbeitsspeicher**: 1 GB bis 128 GB in 1-GB-Schritten

Die Konfiguration wird im Detailbereich der VM angepasst. Änderungen an CPU und RAM erfordern einen Neustart der VM.

Festplatte

Die Festplattengröße wird bei Erstellung der VM festgelegt. Disk-Images werden als Sparse-Dateien auf APFS angelegt und belegen nur den tatsächlich genutzten Speicherplatz.

Netzwerk

- **NAT** (Standard): Die VM erhält eine private IP-Adresse. Der gesamte ausgehende Datenverkehr wird über den Host geroutet.
- **Bridged**: Noch nicht verfügbar. Der Modus setzt eine Freigabe durch Apple voraus.

Optional kann eine eigene MAC-Adresse pro Netzwerk-Interface vergeben werden.

Display

Preset	Auflösung	DPI
HD	1280×720	100
Full HD	1920×1080	144
WUXGA	1920×1200	144
QHD	2560×1440	144
4K	3840×2160	218

Zusätzlich kann eine beliebige Auflösung zwischen 640×480 und 7680×4320 mit DPI zwischen 72 und 326 eingestellt werden.

Audio

- **Audioausgabe**: Standardmäßig aktiviert
- **Audioeingabe**: Standardmäßig deaktiviert, per Toggle aktivierbar

Pro VM kann ein bevorzugtes Ein- und Ausgabegerät zugewiesen werden.

Für Zwischenablage-Sharing über das Audio-Subsystem benötigen Linux-VMs das Paket `spice-vdagent`.

Shared Folders (VirtioFS)

Ordner vom Host können in die VM eingebunden werden:

- **Tag**: Bezeichner für den Mount-Punkt in der VM
- **Host-Pfad**: Ordner auf dem Mac
- **Nur Lesen**: Schreibzugriff aus der VM verhindern
- **Automount**: Ordner beim VM-Start automatisch mounten (Standard: aktiviert)

Rosetta 2

Für Linux-VMs auf Apple Silicon lässt sich Rosetta 2 aktivieren. Damit können x86_64-Binaries in ARM64-Linux-VMs ausgeführt werden. Diese Option ist standardmäßig deaktiviert und muss pro VM aktiviert werden.

USB-Geräte

Optional kann ein XHCI-USB-Controller aktiviert werden. Disk-Images lassen sich als USB-Massenspeicher einbinden, wahlweise im Nur-Lesen-Modus.

VM-Verwaltung

Starten und Stoppen

- ⌘**R** — VM starten
- ⌘**.** — VM stoppen (ACPI-Shutdown)
- ⌘⇧**P** — Pause / Fortsetzen

Das Stoppen erfolgt über ein ACPI-Shutdown-Signal an das Gastsystem. Reagiert das Gastsystem nicht, kann über die API ein Force-Stop ausgelöst werden.

Inline-Display und Fullscreen

Die VM-Ausgabe wird direkt im Hauptfenster angezeigt (Inline-Display). Mit **Ctrl+⌘F** wechselt die Darstellung in den Vollbildmodus. **Escape** kehrt zum Fenster zurück.

Auto-Start

Pro VM lässt sich konfigurieren, ob sie beim App-Start automatisch gestartet werden soll. Die Einstellung befindet sich in den VM-Details.

Zwischenablage

Die Zwischenablage wird standardmäßig zwischen Host und VM geteilt. Unter Linux wird dafür `spice-vdagent` benötigt, macOS-VMs unterstützen dies nativ.

Notizen

Jede VM verfügt über ein Textfeld für eigene Notizen — etwa für Zugangsdaten, Konfigurationshinweise oder Projektbezüge.

Diagnose

Im Diagnose-Tab stehen folgende Informationen zur Verfügung:

- **Betriebszeit**: Laufzeit seit VM-Start
- **Status**: Aktueller Zustand (Gestoppt, Startend, Laufend, Pausiert, Stoppend, Fehler)

- **Startdauer**: Zeit vom Start bis zum Running-Status
- **Konsolenpuffer**: Aktuelle Größe des seriellen Konsolenausgabe-Puffers (max. 65 KB)

Fehlerhistorie

Die letzten 20 Fehler werden mit Zeitstempel und vorherigem VM-Status gespeichert.

Konsole

- **Auto-Refresh**: Regelmäßige Aktualisierung im konfigurierbaren Intervall (1s / 2s / 5s)
- **Befehlseingabe**: Kommandos direkt an die VM senden
- **Puffer leeren**: Konsolenausgabe zurücksetzen

macOS Auto-Setup

Setup-Assistent überspringen

Nemeton markiert alle Schritte des macOS-Setup-Assistenten als erledigt — darunter Apple-ID, iCloud, Datenschutz, FileVault, Ortungsdienste und mehr. Die VM bootet direkt zum Desktop.

Benutzerkonto

Es wird automatisch ein Administratorkonto angelegt:

Option	Standardwert
Benutzername	admin
Vollständiger Name	Admin
Passwort	(konfigurierbar)
Auto-Login	aktiviert

Das Konto wird mit UID 501 in der Gruppe `admin` erstellt.

Auto-Login

Nemeton richtet automatisches Anmelden ein, sodass die VM nach dem Boot direkt den Desktop anzeigt — ohne Anmeldebildschirm.

Erststart-Aktionen

Beim ersten Boot der VM führt ein LaunchDaemon optionale Aktionen aus:

- **SSH aktivieren**: Remote-Zugriff auf die VM per SSH
- **Dock aufräumen**: Standard-Apps aus dem Dock entfernen, Symbolgröße auf 48 Pixel setzen

Der LaunchDaemon entfernt sich nach der Ausführung selbst.

Konfigurationsoptionen

Option	Standard	Beschreibung
Aktiviert	Ja	Master-Schalter für das Auto-Setup
Setup-Assistent überspringen	Ja	Alle Setup-Schritte als erledigt markieren
Auto-Login	Ja	Automatische Anmeldung ohne Passwortabfrage
SSH aktivieren	Nein	Remote-Login per SSH
Dock aufräumen	Ja	Standard-Apps aus dem Dock entfernen

Snapshots

Copy-on-Write auf APFS

Nemeton erstellt Snapshots als Copy-on-Write-Klone auf APFS. Dadurch sind Snapshots nahezu sofort erstellt und belegen nur minimalen zusätzlichen Speicherplatz.

Ein Snapshot umfasst den vollständigen Festplattenzustand:

- Disk-Image (`.img`)
- EFI-Variablen (`efi_vars.bin`)
- Auxiliary Storage (bei macOS-VMs)

Snapshot erstellen

Die VM muss **gestoppt** sein. Über die Snapshot-Ansicht in den VM-Details wird ein neuer Snapshot mit optionalem Namen und Beschreibung angelegt.

Snapshot wiederherstellen

Beim Wiederherstellen wird der aktuelle Festplattenzustand durch den des Snapshots ersetzt. Nemeton prüft vorher, ob ausreichend Speicherplatz vorhanden ist. Die VM muss gestoppt sein.

Snapshot löschen

Snapshots können einzeln gelöscht werden. Das Snapshot-Verzeichnis und die Metadaten werden entfernt.

Einschränkungen

- Maximal **10 Snapshots** pro VM (konfigurierbar in den Einstellungen, 1–50)
- Alle Snapshot-Operationen erfordern eine **gestoppte VM**
- Snapshots werden unter `~/Library/Application Support/Nemeton/VMs/{VMName}/snapshots/` gespeichert

API & Automatisierung

Verbindung

- **Standardport**: 22100 (konfigurierbar, 1024–65535)
- **Authentifizierung**: Bearer-Token im Header `Authorization: Bearer <token>`
- **API-Dokumentation**: Interaktive Redoc-UI unter `/api/v1/docs`
- **OpenAPI-Spezifikation**: `/api/v1/openapi.json`

REST-API

Die vollständige Liste der Endpunkte steht in der OpenAPI-Dokumentation unter /api/v1/docs. Hier die wichtigsten:

VM-Lifecycle

Methode	Endpunkt	Beschreibung
GET	`/api/v1/vms`	Alle VMs auflisten
GET	`/api/v1/vms/:id`	VM-Details
POST	`/api/v1/vms/:id/start`	VM starten
POST	`/api/v1/vms/:id/stop`	VM stoppen (ACPI)
POST	`/api/v1/vms/:id/force-stop`	VM sofort beenden
POST	`/api/v1/vms/:id/pause`	VM pausieren
POST	`/api/v1/vms/:id/resume`	VM fortsetzen

Snapshots

Methode	Endpunkt	Beschreibung
GET	`/api/v1/vms/:id/snapshots`	Snapshots auflisten
POST	`/api/v1/vms/:id/snapshots`	Snapshot erstellen
POST	`/api/v1/vms/:id/snapshots/:sid/restore`	Wiederherstellen
DELETE	`/api/v1/vms/:id/snapshots/:sid`	Löschen

Konsole

Methode	Endpunkt	Beschreibung
GET	`/api/v1/vms/:id/console`	Ausgabe lesen
POST	`/api/v1/vms/:id/console`	Text senden
POST	`/api/v1/vms/:id/console/execute`	Befehl ausführen
DELETE	`/api/v1/vms/:id/console`	Puffer leeren

Echtzeit-Events

Über `/api/v1/events` können Live-Events per WebSocket oder Server-Sent Events (SSE) empfangen werden:

- `vm.started`, `vm.stopped`, `vm.paused`, `vm.resumed`, `vm.error`
- `snapshot.created`, `snapshot.restored`, `snapshot.deleted`
- `console.output`
- `download.started`, `download.completed`, `download.failed`

Webhooks

- **Event-Filter**: Bestimmte Events auswählen oder alle empfangen
- **HMAC-Secret**: Optionale Signatur-Validierung über `X-Nemeton-Secret`-Header
- **Asynchrone Zustellung**: Nicht-blockierend mit automatischem Retry

MCP (Model Context Protocol)

Über `/mcp` steht ein JSON-RPC-2.0-Interface mit über 200 Werkzeugen bereit. Eine Auswahl:

Tool	Beschreibung
`list_vms`	Alle VMs auflisten
`start_vm` / `stop_vm`	VM starten / stoppen
`send_console`	Befehl an Konsole senden
`read_console`	Konsolenausgabe lesen
`take_screenshot`	Screenshot als Base64-JPEG
`create_snapshot`	Snapshot erstellen
`clone_vm`	VM als CoW-Klon duplizieren
`resize_disk`	Festplatte vergrößern
`get_host_info`	Host-Systeminformationen

KI-Assistent

Überblick

Nemeton bringt einen KI-Assistenten mit. In der App heißt er „KI-Agent“. Sie öffnen ihn über den Knopf **KI-Agent** in der Symbolleiste; den Fortschritt einer Aufgabe zeigt das Agent-Fenster. Der Assistent steuert Ihre VMs mit den Werkzeugen der MCP-Schnittstelle und unterliegt denselben Sperren wie ein MCP-Client.

Im Chat steht dauerhaft der Hinweis, dass Sie mit einer KI chatten.

Anbieter

Den Anbieter wählen Sie unter **Einstellungen** → **KI-Agent** → **Anbieter**:

Anbieter	Was Sie brauchen
Nemeton Cloud	Ein Abo von Nemeton Cloud. Sie schließen es im Web-Store ab, mit dem Kundenkonto, mit dem Nemeton angemeldet ist.
Claude (Anthropic), OpenAI, Google Gemini	Einen eigenen API-Schlüssel. Sie zahlen den Anbieter direkt.
Ollama (lokal)	Kein Abo und keinen Schlüssel. Ollama läuft kostenlos auf diesem Mac.

Modell und Endpoint lassen sich auf derselben Seite ändern.

Einwilligung und Einrichtung

Vor der ersten Nutzung fragt der Assistent nach Ihrer Einwilligung. Ohne sie geht nichts an einen KI-Dienst. Widerrufen Sie die Einwilligung, ist der Assistent sofort gesperrt.

Fehlt noch etwas, etwa die Einwilligung, die Anmeldung, das Abo oder ein Schlüssel, nennt der Chat den nächsten Schritt und bringt Sie mit einem Klick dorthin.

Bestätigungsmodus

Unter **Einstellungen** → **KI-Agent** → **Sicherheit** legen Sie fest, wann der Assistent vorher fragt. Vorgabe ist „Sicher“.

Modus	Verhalten
Sicher	Destruktive Aktionen bestätigen Sie vorher.
Paranoid	Nur Lese-Werkzeuge laufen automatisch, alles andere fragt vorher.
Zuschauen	Wie Paranoid; nur der Grund in der Rückfrage ist ein anderer.
Autonom	Bestätigung nur bei Zugriffen auf andere VMs.

- Zugriffe auf eine andere als die aktive VM bestätigen Sie in jedem Modus.
- Bevor der Assistent Dateien auf diesem Mac liest oder schreibt, fragt er nach. Im Modus „Autonom“ lehnt er solche Zugriffe ab.
- Die Sperren unter **API & MCP** gelten auch für den Assistenten: Aufrufe an eine VM, deren Fernsteuerung gesperrt ist, lehnt er ab, auch lesende. Sind destruktive Aktionen dort ausgeschaltet, lehnt er sie ab, statt zu fragen.

Claude Code in der VM

Überblick

Nemeton startet Claude Code in einer macOS-VM, liest mit und bedient die Sitzung. Jede Sitzung läuft im Gast in einem eigenen Terminal-Fenster. Dort sehen Sie live, was Claude Code tut, und können selbst tippen und Fragen beantworten. Die Funktion ist in der Beta.

Voraussetzungen

- **macOS-VM mit Nemeton Bridge**: Claude Code läuft über Nemeton nur in macOS-VMs, nicht in Linux-VMs.
- **Claude Code in der VM installiert**: Nemeton liefert Claude Code nicht mit und installiert es nicht.
- **Eigenes Claude-Konto**: Claude Code läuft in der VM mit Ihrem eigenen Konto unter den Bedingungen von Anthropic für Claude Code. Nemeton startet und bedient es nur.
- **Fernsteuerung eingeschaltet**: Für MCP und den KI-Assistenten muss in der VM „Diese VM per MCP fernsteuern“ eingeschaltet sein.

In Nemeton lässt sich zusätzlich ein API-Schlüssel oder ein langlebiges OAuth-Token (aus `claude setup-token`) hinterlegen. Nemeton gibt es beim Start in die Umgebung von Claude Code.

Starten und bedienen

Sitzungen starten und steuern Sie über die MCP-Schnittstelle, die REST-API oder den KI-Assistenten. Beim Start geben Sie die Aufgabe und das Arbeitsverzeichnis in der VM an. Je VM laufen bis zu vier Sitzungen nebeneinander, jede mit eigener Aufgabe.

Werkzeug	Zweck
`vm_claude_status`	Zeigt, ob Claude Code in der VM installiert ist, welche Version, und wie viele Sitzungen laufen
`vm_claude_start`	Startet eine Sitzung mit Aufgabe und Arbeitsverzeichnis
`vm_claude_list`	Listet laufende und beendete Sitzungen einer VM mit ihrem Zustand
`vm_claude_read`	Liest den Verlauf einer Sitzung
`vm_claude_snapshot`	Liefert den Bildschirm des Terminals als Text
`vm_claude_send`	Schickt eine weitere Nachricht an Claude Code
`vm_claude_wait`	Wartet, bis sich der Zustand einer Sitzung ändert
`vm_claude_interrupt`	Bricht den laufenden Arbeitsschritt ab
`vm_claude_stop`	Beendet eine Sitzung samt Fenster
`vm_claude_show`	Holt das Terminal-Fenster der Sitzung im Gast nach vorn

Für andere Programme gibt es die Terminal-Werkzeuge `vm_terminal_*`: Terminal-Sitzungen im Gast öffnen, lesen, beschreiben und wieder aufnehmen.

VM steuert VM

Claude Code in einer macOS-VM kann über die Nemeton Bridge per MCP Ihre anderen VMs bedienen. Das schalten Sie in den Einstellungen der VM unter **VM steuert VM** ein; ab Werk ist es aus. Der Host, seine Einstellungen, Dateien und Geräte bleiben dabei gesperrt.

Aufgaben und Protokolle

Schreiben Sie keine Passwörter oder Schlüssel in Aufgaben. Aufgaben gehen wie jede Eingabe an Claude. Aufgabe und Protokoll einer Sitzung speichert Nemeton unbefristet im Ordner der VM auf Ihrem Mac; der Terminal-Text liegt in der VM.

Tastenkürzel

Kürzel	Aktion
⌘N	Neue Linux-VM
⌘⇧N	Neue macOS-VM
⌘O	Profil importieren
⌘R	VM starten
⌘.	VM stoppen
⌘⇧P	Pause / Fortsetzen
Ctrl+⌘F	Vollbildmodus umschalten
Escape	Vollbildmodus verlassen

Häufige Fragen

Kann ich macOS-VMs auf einem Intel-Mac erstellen?

Nein. macOS-VMs erfordern Apple Silicon (M1 oder neuer). Linux-VMs funktionieren auf beiden Architekturen.

Wie viele VMs kann ich gleichzeitig laufen lassen?

macOS-VMs: höchstens zwei gleichzeitig je Mac, das ist eine Grenze von Apple. Linux-VMs: keine feste Grenze; wie viele gleichzeitig laufen, hängt vom Arbeitsspeicher und den CPU-Kernen Ihres Macs ab.

Warum belegt mein Disk-Image weniger Platz als konfiguriert?

Nemeton erstellt Sparse-Disk-Images auf APFS. Diese belegen nur den tatsächlich genutzten Speicherplatz und wachsen bei Bedarf.

Können Snapshots bei laufender VM erstellt werden?

Nein. Die VM muss gestoppt sein, um einen konsistenten Snapshot zu gewährleisten.

Wo werden die VM-Daten gespeichert?

Alle VMs liegen unter ~/Library/Application Support/Nemeton/VMs/. Jede VM hat ein eigenes Verzeichnis mit Disk-Image, EFI-Variablen und Snapshot-Ordner.

Was kostet Nemeton?

In der Beta nichts — kein Abo. Die ersten fünf Tage läuft Nemeton ohne Konto, danach registrieren Sie sich kostenlos mit Ihrer E-Mail-Adresse. Wer sich bis zum 15. Dezember 2026 registriert und die E-Mail-Adresse bestätigt, bekommt die Vollversion 1.x samt Pro für die private Nutzung geschenkt; Firmen nutzen Nemeton im Abo und bekommen das erste Jahr geschenkt, wenn sie sich bis dahin registrieren.

Kontakt & Support

Bei Fragen oder Problemen erreichen Sie den Support unter: info@juergenkoller.software

Jürgen Koller Software GmbH
Wilhelmstr. 5, 74072 Heilbronn

Telefon: +49 7131 9244166
E-Mail: info@juergenkoller.software
Web: juergenkoller.software

Nemeton — Virtuelle Maschinen, nativ auf dem Mac.