Nemeton

User Manual

Version 1.0 Beta — October 2026

Jürgen Koller Software GmbH

Contents

Nemeton

Key Features

Getting Started

Installation · System Requirements · Creating Your First VM

VM Creation

Linux VMs · macOS VMs · Defaults

VM Configuration

CPU and Memory · Disk · Network · Display · Audio · Shared Folders (VirtioFS) · Rosetta 2 · USB Devices

VM Management

Starting and Stopping · Inline Display and Fullscreen · Auto-Start · Clipboard · Notes · Diagnostics

macOS Auto-Setup

Skipping the Setup Assistant · User Account · Auto-Login · First Boot Actions · Configuration Options

Snapshots

Copy-on-Write on APFS · Creating a Snapshot · Restoring a Snapshot · Deleting a Snapshot · Limitations

API & Automation

Connection · REST API · Real-Time Events · Webhooks · MCP (Model Context Protocol)

AI Assistant

Overview · Providers · Consent and Setup · Confirmation Mode

Claude Code in the VM

Overview · Requirements · Starting and Operating · VM Controls VM · Tasks and Logs

Keyboard Shortcuts

FAQ

Can I create macOS VMs on an Intel Mac? · How many VMs can I run simultaneously? · Why does my disk image take up less space than configured? · Can snapshots be created while the VM is running? · Where is the VM data stored? · How much does Nemeton cost?

Nemeton

User Manual — Version 1.0 Beta

Nemeton is a native macOS app for managing virtual machines using Apple's Virtualization.framework. Create, configure and run VMs directly on your Mac — without any third-party hypervisor software. All VMs run natively on Apple Silicon.

Key Features

- **Linux & macOS VMs** — create, configure and run virtual machines
- **9 Linux distributions** — with pre-installed disk images and ISO downloads
- **Snapshots** — instant copy-on-write clones on APFS
- **macOS Auto-Setup** — skip setup assistant, pre-configure user accounts
- **REST API & MCP** — REST API with OpenAPI documentation and over 200 MCP tools for automation and AI agents
- **AI assistant** — with Nemeton Cloud, your own API key or Ollama locally
- **Claude Code in the VM** — start, follow and operate sessions in macOS VMs (beta)
- **Shared Folders** — mount host folders via VirtioFS

Getting Started

Installation

Nemeton is available as a direct download. The downloaded DMG contains the app, which can be dragged to the Applications folder.

System Requirements

- **macOS 15 (Sequoia)** or later
- **Apple Silicon** recommended (required for macOS VMs)
- Sufficient disk space for VM disk images (approx. 64 GB per VM or more)

Creating Your First VM

After launching Nemeton, you'll see an empty sidebar. Use **File** → **New Linux VM** (⌘N) or **File** → **New macOS VM** (⌘⇧N) to create a new virtual machine.

For Linux VMs, choose a distribution from the list. Nemeton offers pre-installed disk images for a quick start as well as ISO downloads for manual installation. macOS VMs automatically download the appropriate IPSW installation image from Apple.

The VM appears in the sidebar and can be configured or started immediately.

VM Creation

Linux VMs

Nemeton supports the following Linux distributions:

Distribution	Version	Architecture
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

A pre-installed disk image is available for Fedora Workstation, which skips the ISO installation process. The VM boots directly into a fully configured system.

Downloads use automatic retry logic with three attempts and exponential backoff for connection issues.

macOS VMs

macOS VMs are only available on Apple Silicon Macs. Nemeton automatically downloads the matching IPSW file from Apple. The version is determined based on the host system — only a macOS version matching or older than the host version can be installed.

Defaults

Setting	Linux	macOS
CPU cores	4	4
Memory	8 GB	8 GB
Disk	64 GB	80 GB
Network	NAT	NAT

VM Configuration

CPU and Memory

- **CPU cores**: 1 up to the number of physical CPU cores on the Mac
- **Memory**: 1 GB to 128 GB in 1 GB steps

Configuration is adjusted in the VM detail view. Changes to CPU and memory require a VM restart.

Disk

The disk size is set when creating the VM. Disk images are created as sparse files on APFS and only consume the space actually used by the VM.

Network

- **NAT** (default): The VM gets a private IP address. All outgoing traffic is routed through the host.
- **Bridged**: Not available yet. This mode requires approval from Apple.

A custom MAC address can optionally be assigned per network interface.

Display

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

A custom resolution between 640×480 and 7680×4320 with DPI between 72 and 326 can also be configured.

Audio

- **Audio output**: Enabled by default
- **Audio input**: Disabled by default, can be toggled on

A preferred input and output device can be assigned per VM.

For clipboard sharing via the audio subsystem, Linux VMs require the `spice-vdagent` package.

Shared Folders (VirtioFS)

Host folders can be mounted inside the VM:

- **Tag**: Identifier for the mount point inside the VM
- **Host path**: Folder on the Mac
- **Read only**: Prevent write access from the VM
- **Automount**: Automatically mount the folder at VM startup (default: enabled)

Rosetta 2

For Linux VMs on Apple Silicon, Rosetta 2 can be enabled. This allows running x86_64 binaries inside ARM64 Linux VMs. This option is disabled by default and must be enabled per VM.

USB Devices

An optional XHCI USB controller can be enabled. Disk images can be attached as USB mass storage devices, optionally in read-only mode.

VM Management

Starting and Stopping

- ⌘**R** — Start VM
- ⌘**.** — Stop VM (ACPI shutdown)
- ⌘⇧**P** — Pause / Resume

Stopping sends an ACPI shutdown signal to the guest system. If the guest does not respond, a force stop can be triggered via the API.

Inline Display and Fullscreen

The VM output is shown directly in the main window (inline display). Press **Ctrl+⌘F** to switch to fullscreen mode. **Escape** returns to the window.

Auto-Start

Each VM can be configured to start automatically when the app launches. The setting is located in the VM details.

Clipboard

The clipboard is shared between host and VM by default. Linux requires `spice-vdagent` for this, macOS VMs support it natively.

Notes

Each VM has a text field for custom notes — for credentials, configuration hints or project references.

Diagnostics

The diagnostics tab provides the following information:

- **Uptime**: Duration since VM was started
- **Status**: Current state (Stopped, Starting, Running, Paused, Stopping, Error)

- **Startup duration**: Time from starting to running state
- **Console buffer size**: Current serial console output buffer size (max 65 KB)

Error History

The last 20 errors are stored with timestamp and previous VM status.

Console

- **Auto-refresh**: Regular updates at configurable intervals (1s / 2s / 5s)
- **Command input**: Send commands directly to the VM
- **Clear buffer**: Reset console output

macOS Auto-Setup

Skipping the Setup Assistant

Nemeton marks all steps of the macOS setup assistant as completed — including Apple ID, iCloud, Privacy, FileVault, Location Services and more. The VM boots directly to the desktop.

User Account

An administrator account is created automatically:

Option	Default
Username	admin
Full name	Admin
Password	(configurable)
Auto-login	enabled

The account is created with UID 501 in the `admin` group.

Auto-Login

Nemeton sets up automatic login so that the VM displays the desktop directly after boot — without a login screen.

First Boot Actions

On the first boot, a LaunchDaemon executes optional actions:

- **Enable SSH**: Remote access to the VM via SSH
- **Clean Dock**: Remove default apps from the Dock, set icon size to 48 pixels

The LaunchDaemon removes itself after execution.

Configuration Options

Option	Default	Description
Enabled	Yes	Master toggle for auto-setup
Skip setup assistant	Yes	Mark all setup steps as completed
Auto-login	Yes	Automatic login without password prompt
Enable SSH	No	Remote login via SSH
Clean Dock	Yes	Remove default apps from the Dock

Snapshots

Copy-on-Write on APFS

Nemeton creates snapshots as copy-on-write clones on APFS. This makes snapshots near-instant and uses minimal additional disk space.

A snapshot captures the complete disk state:

- Disk image (`.img`)
- EFI variables (`efi_vars.bin`)
- Auxiliary storage (for macOS VMs)

Creating a Snapshot

The VM must be **stopped**. In the snapshot view within the VM details, create a new snapshot with an optional name and description.

Restoring a Snapshot

When restoring, the current disk state is replaced with the snapshot's state. Nemeton checks available disk space beforehand. The VM must be stopped.

Deleting a Snapshot

Snapshots can be deleted individually. The snapshot directory and metadata are removed.

Limitations

- Maximum **10 snapshots** per VM (configurable in settings, 1–50)
- All snapshot operations require a **stopped VM**
- Snapshots are stored under `~/Library/Application Support/Nemeton/VMs/{VMName}/snapshots/`

API & Automation

Connection

- **Default port**: 22100 (configurable, 1024–65535)
- **Authentication**: Bearer token in the `Authorization: Bearer <token>` header
- **API documentation**: Interactive Redoc UI at `/api/v1/docs`
- **OpenAPI specification**: `/api/v1/openapi.json`

REST API

The full list of endpoints is in the OpenAPI documentation at /api/v1/docs. Here are the most important ones:

VM Lifecycle

Method	Endpoint	Description
GET	`/api/v1/vms`	List all VMs
GET	`/api/v1/vms/:id`	VM details
POST	`/api/v1/vms/:id/start`	Start VM
POST	`/api/v1/vms/:id/stop`	Stop VM (ACPI)
POST	`/api/v1/vms/:id/force-stop`	Force stop VM
POST	`/api/v1/vms/:id/pause`	Pause VM
POST	`/api/v1/vms/:id/resume`	Resume VM

Snapshots

Method	Endpoint	Description
GET	`/api/v1/vms/:id/snapshots`	List snapshots
POST	`/api/v1/vms/:id/snapshots`	Create snapshot
POST	`/api/v1/vms/:id/snapshots/:sid/restore`	Restore
DELETE	`/api/v1/vms/:id/snapshots/:sid`	Delete

Console

Method	Endpoint	Description
GET	`/api/v1/vms/:id/console`	Read output
POST	`/api/v1/vms/:id/console`	Send text
POST	`/api/v1/vms/:id/console/execute`	Execute command
DELETE	`/api/v1/vms/:id/console`	Clear buffer

Real-Time Events

Live events are available via WebSocket or Server-Sent Events (SSE) at `/api/v1/events`:

- `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**: Choose specific events or receive all
- **HMAC secret**: Optional signature validation via `X-Nemeton-Secret` header
- **Async delivery**: Non-blocking with automatic retry

MCP (Model Context Protocol)

A JSON-RPC 2.0 interface with over 200 tools is available at `/mcp`. A selection:

Tool	Description
`list_vms`	List all VMs
`start_vm` / `stop_vm`	Start / stop VM
`send_console`	Send command to console
`read_console`	Read console output
`take_screenshot`	Screenshot as Base64 JPEG
`create_snapshot`	Create snapshot
`clone_vm`	Duplicate VM as CoW clone
`resize_disk`	Enlarge disk image
`get_host_info`	Host system information

AI Assistant

Overview

Nemeton comes with an AI assistant. In the app it is called "AI Agent". You open it with the **AI Agent** button in the toolbar; the agent window shows the progress of a task. The assistant controls your VMs with the tools of the MCP interface and is subject to the same locks as an MCP client.

The chat permanently shows a notice that you are chatting with an AI.

Providers

Choose the provider under **Settings** → **AI Agent** → **Provider**:

Provider	What you need
Nemeton Cloud	A Nemeton Cloud subscription. You take it out in the web store, with the customer account Nemeton is signed in with.
Claude (Anthropic), OpenAI, Google Gemini	Your own API key. You pay the provider directly.
Ollama (local)	No subscription and no key. Ollama runs free of charge on this Mac.

Model and endpoint can be changed on the same page.

Consent and Setup

Before its first use, the assistant asks for your consent. Without it, nothing is sent to an AI service. If you withdraw your consent, the assistant is blocked immediately.

If something is still missing, such as consent, sign-in, the subscription or a key, the chat names the next step and takes you there with one click.

Confirmation Mode

Under **Settings** → **AI Agent** → **Security** you decide when the assistant asks first. The default is "Safe".

Mode	Behavior
Safe	You confirm destructive actions first.
Paranoid	Only read tools run automatically, everything else asks first.
Watch	Like Paranoid; only the reason given in the prompt differs.
Autonomous	Confirmation only for access to other VMs.

- You confirm access to any VM other than the active one in every mode.
- Before the assistant reads or writes files on this Mac, it asks. In "Autonomous" mode it refuses such access.
- The locks under **API & MCP** also apply to the assistant: it refuses calls to a VM whose remote control is locked, even read-only ones. If destructive actions are switched off there, it refuses them instead of asking.

Claude Code in the VM

Overview

Nemeton starts Claude Code in a macOS VM, follows along and operates the session. Each session runs in its own Terminal window in the guest. There you see live what Claude Code is doing, and you can type and answer questions yourself. The feature is in beta.

Requirements

- **macOS VM with Nemeton Bridge**: through Nemeton, Claude Code runs only in macOS VMs, not in Linux VMs.
- **Claude Code installed in the VM**: Nemeton does not ship or install Claude Code.
- **Your own Claude account**: Claude Code runs in the VM with your own account under Anthropic's terms for Claude Code. Nemeton only starts and operates it.
- **Remote control switched on**: for MCP and the AI assistant, "Remote-control this VM via MCP" must be switched on for the VM.

You can also store an API key or a long-lived OAuth token (from `claude setup-token`) in Nemeton. Nemeton passes it into Claude Code's environment at start.

Starting and Operating

You start and control sessions through the MCP interface, the REST API or the AI assistant. At start you specify the task and the working directory in the VM. Up to four sessions run side by side per VM, each with its own task.

Tool	Purpose
`vm_claude_status`	Shows whether Claude Code is installed in the VM, which version, and how many sessions are running
`vm_claude_start`	Starts a session with a task and working directory
`vm_claude_list`	Lists running and ended sessions of a VM with their state
`vm_claude_read`	Reads the history of a session
`vm_claude_snapshot`	Returns the terminal screen as text
`vm_claude_send`	Sends another message to Claude Code
`vm_claude_wait`	Waits until the state of a session changes
`vm_claude_interrupt`	Interrupts the current step
`vm_claude_stop`	Ends a session including its window
`vm_claude_show`	Brings the session's Terminal window to the front in the guest

For other programs there are the terminal tools `vm_terminal_*`: open, read, write and resume terminal sessions in the guest.

VM Controls VM

Claude Code in a macOS VM can operate your other VMs via MCP through the Nemeton Bridge. You switch this on in the VM's settings under **VM controls VM**; it is off by default. The host, its settings, files and devices stay locked.

Tasks and Logs

Don't put passwords or keys in tasks. Tasks go to Claude like any other input. Nemeton stores the task and log of a session indefinitely in the VM's folder on your Mac; the terminal text stays in the VM.

Keyboard Shortcuts

Shortcut	Action
⌘N	New Linux VM
⌘⇧N	New macOS VM
⌘O	Import profile
⌘R	Start VM
⌘.	Stop VM
⌘⇧P	Pause / Resume
Ctrl+⌘F	Toggle fullscreen
Escape	Exit fullscreen

FAQ

Can I create macOS VMs on an Intel Mac?

No. macOS VMs require Apple Silicon (M1 or later). Linux VMs work on both architectures.

How many VMs can I run simultaneously?

macOS VMs: at most two at a time per Mac, a limit set by Apple. Linux VMs: no fixed limit; how many run at once depends on the memory and CPU cores of your Mac.

Why does my disk image take up less space than configured?

Nemeton creates sparse disk images on APFS. These only occupy the space actually used and grow as needed.

Can snapshots be created while the VM is running?

No. The VM must be stopped to ensure a consistent snapshot.

Where is the VM data stored?

All VMs are located under ~/Library/Application Support/Nemeton/VMs/. Each VM has its own directory with disk image, EFI variables and snapshot folder.

How much does Nemeton cost?

Nothing during the beta — no subscription. For the first five days Nemeton runs without an account; after that you register for free with your email address. Register by December 15, 2026, confirm your email address, and the full version 1.x, Pro included, is yours for free for personal use; businesses use Nemeton with a subscription and get the first year for free if they register by then.

Contact & Support

For questions or issues, please contact support at: info@juergenkoller.software

Jürgen Koller Software GmbH
Wilhelmstr. 5, 74072 Heilbronn

Phone: +49 7131 9244166
Email: info@juergenkoller.software
Web: juergenkoller.software

Nemeton — Virtual machines, native on your Mac.