Nemeton

User Manual

Version 0.2 — April 2026

Jürgen Koller Software GmbH

Contents

Introduction

Nemeton is a native macOS app for managing virtual machines using Apple's Virtualization.framework. It lets you create, configure and run Linux and macOS VMs directly on your Mac — without any third-party hypervisor software.

All VMs run natively on Apple Silicon and leverage the processor's hardware virtualization capabilities. macOS VMs require an Apple Silicon Mac. Linux VMs run on both architectures.

System Requirements

- **macOS 14 (Sonoma)** or later
- **Apple Silicon** recommended (required for macOS VMs)
- Sufficient disk space for VM disk images (approx. 64 GB per VM or more)

Getting Started

Installation

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

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. All other distributions require downloading the ISO and manual installation.

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.

The default configuration for macOS VMs includes 4 CPU cores, 8 GB RAM, 80 GB disk and a display resolution of 1920×1200.

Defaults

New VMs are created with the following presets:

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

Nemeton offers two network modes:

- **NAT** (default): The VM gets a private IP address. All outgoing traffic is routed through the host.
- **Bridged**: The VM connects directly to the host's network and receives its own IP address via DHCP.

A custom MAC address can optionally be assigned per network interface. If omitted, one is generated automatically.

Display

Nemeton provides predefined resolutions:

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)

Mounting uses the VirtioFS protocol. The VM must have the appropriate driver or guest support installed.

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.

macOS VMs use Rosetta 2 natively through the host system.

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

VMs are controlled via the toolbar or keyboard shortcuts:

- **⌘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. This helps track recurring issues.

Console

The serial console shows real-time VM output:

- **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

When installing a macOS VM, Nemeton can automatically skip the setup assistant and pre-configure a user account.

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. Password hashes are generated locally using SALTED-SHA512-PBKDF2.

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

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

What Is Saved?

A snapshot captures the complete disk state of the VM:

- 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

In the debug build, Nemeton starts an HTTP server for automation and integration. The server provides a REST API, WebSocket/SSE events, webhooks and an MCP interface.

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`

The token is generated on first launch and stored in the macOS Keychain. It can be viewed and copied in the settings.

REST API

The API provides 42 endpoints across the following areas:

VM Lifecycle

Method	Endpoint	Description
GET	`/api/v1/vms`	List all VMs with status
GET	`/api/v1/vms/:id`	Get 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 snapshot
DELETE	`/api/v1/vms/:id/snapshots/:sid`	Delete snapshot

Console

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

Additional Endpoints

Method	Endpoint	Description
GET	`/api/v1/vms/:id/runtime`	Runtime metrics
GET	`/api/v1/vms/:id/screenshot`	Screenshot as JPEG
GET	`/api/v1/distros`	Available Linux distributions
POST	`/api/v1/profiles`	Create VM profile
PATCH	`/api/v1/profiles/:id`	Update VM profile
GET	`/api/v1/webhooks`	List webhooks
POST	`/api/v1/webhooks`	Register webhook

Real-Time Events

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

Available events:

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

Webhooks

External URLs can be notified on VM events:

- **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 24 tools is available at `/mcp`. This allows integrating Nemeton with AI assistants like Claude.

Key MCP tools:

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

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?

There is no fixed limit. The number depends on the available resources (CPU cores, RAM, disk space) 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 by the VM and grow as needed up to the configured size.

Can snapshots be created while the VM is running?

No. The VM must be stopped to ensure a consistent snapshot. Paused VMs are also not supported.

Where is the VM data stored?

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

How do I activate my license?

For the direct download version, enter the license key in the settings under **License**. For the App Store version, activation happens automatically through the purchase.

After installation, a 5-day trial period with full functionality begins.

Support & Contact

Jürgen Koller Software GmbH

Email: support@juergenkoller.software Web: https://store.juergenkoller.software/apps/nemeton

---