@serve.zone/moxytool

a cli tool designed for management of proxmox environments

- readme.md for @serve.zone/moxytool
- changelog.md for @serve.zone/moxytool

readme.md for @serve.zone/moxytool

> Proxmox Administration Tool for vGPU setup, VM management, and cluster configuration

npm version License: MIT

Overview

MOXYTOOL is a comprehensive command-line tool for managing Proxmox servers, with a focus on simplified vGPU setup and advanced server configuration. Built with Deno and compiled to native binaries for maximum performance and portability.

Features

- **vGPU Setup**: Automated installation and configuration of NVIDIA vGPU support on Proxmox
- **Cross-Platform**: Native binaries for Linux, macOS, and Windows
- **Multi-Architecture**: Support for x64 and ARM64 processors
- **Interactive CLI**: User-friendly command-line interface with detailed guidance
- **Proxmox Integration**: Deep integration with Proxmox VE for seamless management

Installation

One-Line Installation (Recommended)

```
# Download and install MOXYTOOL automatically
curl -sSL https://code.foss.global/serve.zone/moxytool/raw/branch/main/install.sh | sudo bash
```

This will:

- Detect your platform automatically (Linux x64/ARM64, macOS Intel/Apple Silicon, Windows)
- Download the latest binary from Gitea releases (~400-500KB)
- Install to `/usr/local/bin/moxytool`
- Make it available system-wide

Via npm (Alternative)

Install globally using npm:

```
npm install -g @serve.zone/moxytool
```

or with pnpm:

```
pnpm install -g @serve.zone/moxytool
```

Benefits:

- Automatic platform detection and binary download
- Easy updates via `npm update -g @serve.zone/moxytool`
- Version management with npm
- Works with Node.js >=14

Usage

Updating MOXYTOOL

Update to the latest version from the repository:

```
moxytool update
```

This command will:

- Check the current version
- Fetch the latest release from Gitea
- Automatically download and install the update if available
- Preserve your existing configuration

vGPU Setup

Install and configure NVIDIA vGPU support on your Proxmox host:

```
sudo moxytool vgpu-setup
```

Arguments

- `--step <number>` - Force execution at a specific installation step
- `--url <url>` - Use a custom driver URL (.run or .zip format)
- `--file <path>` - Use a local driver file
- `--debug` - Enable debug output mode

Examples

```
# Basic setup with interactive prompts
sudo moxytool vgpu-setup

# Use a custom driver URL
sudo moxytool vgpu-setup --url https://example.com/driver.run

# Use a local driver file
sudo moxytool vgpu-setup --file /path/to/driver.run

# Resume at a specific step
sudo moxytool vgpu-setup --step 2

# Debug mode
sudo moxytool vgpu-setup --debug
```

Installation Process

1. **Prerequisites**: Ensure virtualization is enabled in BIOS (Intel Vt-d or AMD IOMMU)
2. **Run Setup**: Execute `sudo moxytool vgpu-setup`
3. **Follow Prompts**: The installer will guide you through the process
4. **Reboot**: System will require a reboot after initial setup
5. **Complete Setup**: Run the command again after reboot to finish installation
6. **Verify**: Check installation with `mdevctl types`

Post-Installation

After successful installation:

1. **Verify vGPU profiles**: `mdevctl types`
2. **Configure VMs**: Add vGPU devices in Proxmox web UI (VM → Hardware → Add → PCI Device)
3. **Install guest drivers**: Download and install NVIDIA vGPU guest drivers in your VMs

Community Scripts

Access and deploy 400+ community-maintained Proxmox installation scripts:

```
# List all available scripts
moxytool scripts list

# Search for specific applications
moxytool scripts search docker
moxytool scripts search homeassistant

# View detailed information
moxytool scripts info docker

# Install a script
sudo moxytool scripts run docker

# Refresh the script index
moxytool scripts refresh
```

Features:

- Automatic daily index updates (cached locally)
- 400+ LXC containers and VM templates
- Full interactive installation support
- Applications include: Docker, Jellyfin, Home Assistant, Pi-hole, Nextcloud, and many more

Script Categories:

- Containerization (Docker, Podman, Kubernetes)
- Media servers (Plex, Jellyfin, Emby)
- Home automation (Home Assistant, Node-RED)

- Development tools (GitLab, Jenkins, Gitea)
- Network tools (Pi-hole, AdGuard, WireGuard)
- Databases (PostgreSQL, MariaDB, MongoDB)
- And much more...

Requirements

- Proxmox VE 7.4-9.x
- Root/sudo access
- Internet connection for downloading scripts/drivers

Note: The tool comes as a pre-compiled binary - no runtime dependencies needed!

Supported Platforms

- **Linux**: x64, ARM64
- **macOS**: x64, ARM64 (Apple Silicon)
- **Windows**: x64

Development

Note: Development requires Deno. End users don't need Deno - they use pre-compiled binaries.

Prerequisites

- Deno 2.x or later
- Bash (for compilation scripts)

Building from Source

```
# Clone the repository
git clone https://code.foss.global/serve.zone/moxytool.git
cd moxytool

# Run locally with Deno
```

```
deno task dev

# Compile binaries for all platforms
deno task compile:all

# Run tests
deno task test
```

Project Structure

```
moxytool/
├── mod.ts                      # Main entry point
├── deno.json                   # Deno configuration
├── package.json                # NPM package manifest
├── ts/                         # TypeScript source files
│   ├── moxytool.cli.ts        # CLI command definitions
│   ├── moxytool.plugins.ts    # Plugin imports
│   ├── moxytool.logging.ts    # Logging setup
│   ├── moxytool.paths.ts      # Path definitions
│   └── index.ts               # Node.js entry point
├── bin/                        # Binary wrapper
│   └── moxytool-wrapper.js    # NPM binary wrapper
├── scripts/                    # Build scripts
│   ├── compile-all.sh         # Compilation script
│   └── install-binary.js      # Binary installation
└── dist/                       # Compiled binaries
    └── binaries/
```

Credits

MOXYTOOL uses the excellent [proxmox-vgpu-installer] by anomixer for the core vGPU installation process, which supports Proxmox v9.

License

MIT License - see LICENSE file for details

Support

- **Issues**: https://code.foss.global/serve.zone/moxytool/issues
- **Repository**: https://code.foss.global/serve.zone/moxytool

Related Projects

- NUPST - Network UPS Shutdown Tool
- SPARK - Server Configuration and Management Tool

Made with ❤ by Serve Zone

changelog.md for @serve.zone/moxytool

2025-10-29 - 1.5.1 - fix(scriptindex)

Improve script search: use ObjectSorter with weighted results prioritizing slug and name

- Replaced smartfuzzy.FuzzyMatcher with smartfuzzy.ObjectSorter for multi-field fuzzy searching
- Search now runs across slug, name, and description fields
- Added weighting so slug matches are prioritized, name matches receive a medium boost
- Search returns ordered script objects based on adjusted score

2025-10-29 - 1.5.0 - feat(scripts)

Add fuzzy search and type filtering for community scripts; improve scripts CLI output and cache handling

- Integrate @push.rocks/smartfuzzy and use FuzzyMatcher to provide ranked, fuzzy search results for scripts
- Add optional type filtering to scripts search (e.g. type:vm, type:ct, type:pve) and parse a --filter option in the CLI
- Improve scripts CLI output: colored type badges, truncated descriptions, clearer usage/help text and filtered result messaging
- Optimize ScriptIndex.loadCache to avoid reloading when cache is already present
- Update deno.json to include the smartfuzzy dependency and export/import smartfuzzy in plugins

2025-10-28 - 1.4.2 - fix(scriptindex)

Handle missing script metadata fields in ScriptIndex.search to prevent crashes

- Add null/undefined checks for name, slug, and description in ScriptIndex.search to avoid runtime exceptions when script metadata is incomplete
- Improves robustness of scripts search against partially populated or malformed cached metadata

2025-10-28 - 1.4.1 - fix(cli)

Fallback to 'unknown' when script.slug is missing in scripts list

- Fixes a potential runtime error when listing scripts if a script entry lacks a slug
- Uses a safe fallback ('unknown') before calling padEnd to ensure stable output
- Modified file: ts/moxytool.cli.ts

2025-10-28 - 1.4.0 - feat(cli)

Improve CLI output and logging with colored header, grouped script listings, and ANSI-styled logger

- Set smartcli instance version from deno.json to surface the package version in the CLI
- Revamp standard command output with a colored ASCII header, clearer commands list, and improved usage line
- Group script index output by type including Proxmox VE host (pve), Containers (ct), Virtual Machines (vm), and Other
- Enhance scripts listing formatting (slug padding and bullet points) for readability
- Replace timestamped logger messages with ANSI-colored output and icons for error/warn/success/info

2025-10-28 - 1.3.6 - fix(deps)

Bump smartcli dependency and add local settings file

- Bumped @push.rocks/smartcli from ^4.0.18 to ^4.0.19 in deno.json
- Added .claude/settings.local.json (development/local settings file)

2025-10-28 - 1.3.5 - fix(smartcli)

Bump @push.rocks/smartcli to ^4.0.18 and add local settings file for tooling permissions

- Updated dependency @push.rocks/smartcli from ^4.0.16 to ^4.0.18 in deno.json
- Added a local settings file (.claude/settings.local.json) to configure runtime/tooling permissions (web fetch domains, bash/deno/npm command allowances, and local read access)
- No code API changes; this is a dependency/infra update — incrementing patch version

2025-10-28 - 1.3.4 - fix(smartcli)

Update @push.rocks/smartcli to ^4.0.16 and add local Claude settings

- Bump dependency in deno.json: @push.rocks/smartcli from ^4.0.15 to ^4.0.16
- Add .claude/settings.local.json containing local permissions/configuration (development/local-only file)

2025-10-28 - 1.3.3 - fix(deno.json)

Bump @push.rocks/smartcli to ^4.0.15 and add local Claude settings

- Updated deno.json: @push.rocks/smartcli ^4.0.14 → ^4.0.15
- Added .claude/settings.local.json with local permissions for development/CI
- No runtime source changes; dependency update only — recommend a patch release

2025-10-28 - 1.3.2 - fix(cli)

Correct scripts subcommand argument parsing and bump smartcli dependency

- Fix scripts command argument indices so the subcommand is read from argvArg.*[1] and subsequent arguments from argvArg.*[2]. This resolves incorrect handling of 'scripts search', 'scripts info' and 'scripts run' inputs.

- Upgrade @push.rocks/smartcli dependency from ^4.0.11 to ^4.0.14 in deno.json for compatibility/stability improvements.

2025-10-27 - 1.3.1 - fix(publish)

Switch publish registry to internal Verdaccio instance and add local CI settings

- Update package.json publishConfig.registry from https://registry.npmjs.org/ to https://verdaccio.lossless.digital/ to publish packages to the internal Verdaccio registry.
- Add .claude/settings.local.json to include local CI/dev settings (local configuration only).

2025-10-27 - 1.3.0 - feat(cli)

Add automatic update command and documentation updates

- Add 'update' CLI command that checks the latest Gitea release and runs the install script to perform a self-update
- Implements release fetch/compare logic and executes the repository install.sh via curl
- Update README to document the one-line installer and the new 'moxytool update' usage
- Update changelog to note the new update command and installation clarifications

2025-10-27 - 1.2.0 - feat(scripts)

Add community scripts subsystem: script index, runner, and CLI commands with background refresh; update docs and paths

- New `scripts` command with subcommands: list, search, info, run, refresh (implemented in ts/moxytool.cli.ts)
- Added ScriptIndex (ts/moxytool.classes.scriptindex.ts) to fetch and cache ~400 community scripts with a 24h TTL and background refresh
- Added ScriptRunner (ts/moxytool.classes.scriptrunner.ts) to execute community installation scripts interactively via bash/curl
- Background index refresh at startup and explicit refresh command; cache saved under /etc/moxytool/scripts
- README and changelog updated with scripts usage and features; Proxmox support range updated to 7.4-9.x
- Updated module exports in mod.ts and minor logging change in ts/index.ts
- Added script-related paths (scriptsCacheDir, scriptsIndexFile) to ts/moxytool.paths.ts

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog, and this project adheres to Semantic Versioning.

1.1.0 - 2025-01-27

Added

- `update` command for automatic self-updating from Gitea releases
- `scripts` command for Proxmox community scripts management
- Access to 400+ community-maintained installation scripts
- Automatic daily index updates with local caching
- Script search and filtering capabilities
- Interactive script execution with full stdin/stdout/stderr passthrough
- Support for both LXC containers and VM templates
- Script metadata display (requirements, ports, credentials)
- One-line installation script as primary installation method

Features

- `moxytool update` - Update MOXYTOOL to the latest version automatically
- `moxytool scripts list` - List all available scripts
- `moxytool scripts search <query>` - Search scripts by keyword
- `moxytool scripts info <slug>` - View detailed script information
- `moxytool scripts run <slug>` - Execute installation scripts
- `moxytool scripts refresh` - Force update the script index

Changed

- Updated Proxmox version support to 7.4-9.x (from 7.4-8.x)
- Updated vGPU installer to anomixer fork with Proxmox v9 support

1.0.0 - 2025-01-24

Added

- Initial release of MOXYTOOL
- `vgpu-setup` command for automated Proxmox vGPU installation
- Support for NVIDIA vGPU on Proxmox hosts
- Interactive installer integration with wvthoog/proxmox-vgpu-installer
- Cross-platform binary support (Linux, macOS, Windows)
- Multi-architecture support (x64, arm64)

Features

- Automated vGPU driver download and installation
- Support for custom driver URLs and local files
- Debug mode for troubleshooting
- Step-by-step installation process
- Verification of Proxmox installation before setup