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

Revision #3
Created 2026-03-28 11:14:35 UTC by foss.global Team
Updated 2026-03-28 12:21:21 UTC by foss.global Team