PUQ Mautic Image not found or type unknown

Virtual Machine Templates

Proxmox KVM module **WHMCS**

Order now | Download | FAQ

Overview

The module provisions virtual machines by cloning existing Proxmox VM templates. Before using the module, you need to prepare at least one VM template in your Proxmox environment. Templates must be properly prepared inside the Proxmox panel — the module does not install an operating system from ISO, it only clones a ready template and applies configuration via cloud-init.

Physical Requirements

VM templates must meet the following specifications:

- **Resource sizing**: all template parameters (CPU cores, RAM, system disk size) must be **smaller** than the smallest package you plan to offer clients. The module can grow disks and increase CPU/RAM during deployment, but it **cannot shrink** them.
- **Multi-disk layout**: if you plan to offer VMs with multiple disks on different storage locations, create those disks on the appropriate storage already in the template. Otherwise Proxmox may consolidate all disks on the same storage during clone.
- **Cloud-init drive**: a cloud-init disk is mandatory for automatic VM configuration after cloning.
- **Partition layout**: partitions of the system disk must be arranged so that the **root partition is the LAST** partition in the table. This is required for automatic root filesystem expansion by `cloud-initramfs-growroot` during first boot.

Cloud-Init Requirement

Cloud-init is **required** for automatic VM configuration. The module uses cloud-init to set:

- Hostname
- IP address and network configuration
- DNS servers
- User account and password
- SSH keys

Without cloud-init, the module cannot automatically configure the VM's network and credentials after cloning.

Creating a Template

Step 1: Create a Base VM

Create a new VM in Proxmox with your desired operating system. When partitioning the system disk, ensure the **root partition is the LAST** one in the partition table so that it can be automatically expanded on first boot.

Step 2: Enable Root SSH Access

The module uses the `root` user to push cloud-init configuration and manage the VM. Enable root login via SSH inside the template:

```
# /etc/ssh/sshd_config
PermitRootLogin yes
PasswordAuthentication yes

systemctl restart sshd
```

Step 3: Install Cloud-Init

Install cloud-init together with the growroot and utility packages inside the VM:

```
# Debian/Ubuntu
apt update && apt install -y cloud-initramfs-growroot cloud-init cloud-utils

# CentOS/RHEL/AlmaLinux
```

```
yum install -y cloud-init cloud-utils-growpart

# openSUSE
zypper install cloud-init growpart
```

> `cloud-initramfs-growroot` / `cloud-utils-growpart` is what actually expands the root partition to fill the resized disk during deployment. Without it the client VM will boot with the original template disk size.

Step 4: Enable Cloud-Init Services

Cloud-init is split into four systemd services — all of them must be enabled so the VM picks up configuration on every boot:

```
systemctl enable cloud-init-local.service
systemctl enable cloud-init.service
systemctl enable cloud-config.service
systemctl enable cloud-final.service
```

Step 5: Clean Up the VM

Before converting to a template, clean up the VM so that every clone starts fresh:

```
# Remove default users created during OS install (ubuntu, debian, centos, etc.)
# The module creates the user defined in the product configuration.
userdel -r ubuntu 2>/dev/null || true
userdel -r debian 2>/dev/null || true

# Remove SSH host keys (they will be regenerated on first boot)
rm -f /etc/ssh/ssh_host_*

# Clean cloud-init state
cloud-init clean --logs

# Remove machine ID (will be regenerated)
truncate -s 0 /etc/machine-id
rm -f /var/lib/dbus/machine-id
```

```
ln -s /etc/machine-id /var/lib/dbus/machine-id

# Clear logs
find /var/log -type f -exec truncate -s 0 {} \;

# Clear bash history
cat /dev/null > ~/.bash_history && history -c
```

Step 6: Add Cloud-Init Drive

In the Proxmox web UI:

1. Select the VM
2. Go to **Hardware**
3. Click **Add > CloudInit Drive**
4. Select the storage for the cloud-init drive

Step 7: Convert to Template

In the Proxmox web UI:

1. Right-click the VM
2. Select **Convert to Template**

Template Configuration Tips

- **Use a unique VMID** for each template to avoid conflicts
- **Keep templates on shared storage** if you have a multi-node cluster, or use local storage with migration enabled in the module settings
- **Install the QEMU Guest Agent** (`qemu-guest-agent` package) for improved VM management and status reporting
- **Configure serial console** if you want noVNC console access to work properly
- **Minimize the template disk size** — the module can resize disks during deployment, but it cannot shrink them
- **Test the template** by manually cloning it and verifying that cloud-init applies the configuration correctly

Pre-built Templates

If you don't want to build templates from scratch, PUQcloud publishes two separate sets of ready-made VM templates as Proxmox backup archives (VMA format, `.vma.zst`):

1. **Prebuild Proxmox OS templates** — custom minimal installations built by PUQcloud from scratch.
2. **Official cloud images with root access** — upstream cloud images (Debian Cloud, Ubuntu Cloud, CentOS GenericCloud) modified so that root SSH login works out of the box.

Both sets are hosted under the same root folder:

- https://files.puqcloud.com/Proxmox_OS_Templates/

Download Prebuild Proxmox OS Templates

Custom PUQcloud builds — 5 GB `virtio` disk, no swap, minimal install, root SSH enabled, default password `puqcloud`, timezone Europe/Warsaw.

Debian

- Debian 10
- Debian 11
- Debian 12

Ubuntu

- Ubuntu 18
- Ubuntu 20
- Ubuntu 22

CentOS

- CentOS 7
- CentOS 8
- CentOS 9

Proxmox

- PBS 2.2

Official Cloud Images with Root Access

These are the **upstream cloud images** from Debian, Ubuntu and CentOS, modified by PUQcloud so that root SSH login is enabled out of the box. Use them if you prefer the official distribution builds over custom ones.

> Stock upstream cloud images disable root SSH login by default and create a non-root user (`debian`, `ubuntu`, `centos`). The module requires the `root` account to push cloud-init configuration and run management commands — that's why the "with root access" variants exist.

Debian

- Debian 10 — ~245 MB
- Debian 11 — ~275 MB
- Debian 12 — ~298 MB
- Debian 13 — ~307 MB

Ubuntu

- Ubuntu 20.04 — ~893 MB
- Ubuntu 22.04 — ~626 MB
- Ubuntu 23.10 — ~731 MB

CentOS

- CentOS 7 — ~1.01 GB
- CentOS 9 — ~1.10 GB

> File names may change over time. If a direct link returns 404, open the parent folder on files.puqcloud.com and grab the latest archive for the OS version you

> need.

Importing a Pre-built Template

1. Copy the `.vma.zst` file to your Proxmox node, for example into `/var/lib/vz/dump/`:

   ```
   cd /var/lib/vz/dump/
   wget https://files.puqcloud.com/Proxmox_OS_Templates/Debian/Debian-12/vzdump-qemu-1012-2024_03_23-17_11_11.vma.zst
   ```

2. Restore the backup to a new VMID (pick a free ID, e.g. `9012`) and target storage (e.g. `local-lvm`):

   ```
   qmrestore /var/lib/vz/dump/vzdump-qemu-1012-2024_03_23-17_11_11.vma.zst 9012 --storage local-lvm
   ```

   Alternatively, use the Proxmox web UI: **Datacenter > Storage > Backups > Restore**.
3. Open the restored VM, change the default root password (`puqcloud` for the prebuild set), verify the cloud-init drive is present, then right-click the VM and choose **Convert to Template**.

> **Disclaimer:** Your use of these operating systems is at your own risk. PUQcloud does not guarantee the correct operation or security of the pre-built templates or the root-access cloud images. Always review, update and harden them before offering to clients.

Supported Operating Systems

The module supports any operating system that Proxmox can run as a KVM virtual machine, provided it has working cloud-init (or cloudbase-init on Windows). Tested and known to work:

- **Linux**: Debian, Ubuntu, CentOS, AlmaLinux, Rocky Linux, openSUSE, Proxmox Backup Server
- **Windows**: Server / Desktop editions with `cloudbase-init` installed

Configuring Templates in WHMCS

Templates are selected in the product configuration under the **Module Settings** tab. You can also

offer multiple templates to clients via Configurable Options, allowing them to choose their preferred operating system during order.

Revision #7
Created 10 April 2026 12:07:31 by Ruslan
Updated 15 May 2026 07:07:07 by Ruslan