

Ubuntu 22.04 Template with Cloud-Init

Creating a cloud-init template is relatively simple and straightforward to do. Essentially, you'll want to create your VM with base configurations, and then convert it to a template within Proxmox VE. Follow the steps below to create your own template with cloud-init.

Create your VM in Proxmox

The screenshot shows the 'Create: Virtual Machine' dialog box in Proxmox VE, with the 'General' tab selected. The dialog has a dark theme and a close button in the top right corner. The tabs at the top are General, OS, System, Disks, CPU, Memory, Network, and Confirm. The 'General' tab contains the following fields:

- Node:** A dropdown menu showing 'pve'.
- Resource Pool:** A dropdown menu.
- VM ID:** A dropdown menu showing '901'.
- Name:** A text input field containing 'ubuntu-22.04-server-template'.
- Start at boot:** A checkbox that is currently unchecked.
- Start/Shutdown order:** A dropdown menu showing 'any'.
- Startup delay:** A dropdown menu showing 'default'.
- Shutdown timeout:** A dropdown menu showing 'default'.
- Tags:** A section with the text 'No Tags' and a '+' button to add tags.

At the bottom of the dialog, there is a 'Help' button with a question mark icon, an 'Advanced' checkbox which is checked, and 'Back' and 'Next' buttons.

- Assign a VM ID, I keep my templates in the 900s range, and then give your VM a name

OS Settings

Create: Virtual Machine ✕

General **OS** System Disks CPU Memory Network Confirm

☐ Use CD/DVD disc image file (iso)

Storage: local ✓

ISO image: ✓

Guest OS:

Type: Linux ✓

Version: 6.x - 2.6 Kernel ✓

☐ Use physical CD/DVD Drive

☒ Do not use any media

Advanced ☒ Back Next

- Select "Do not use any media"
 - We'll be adding cloud-init drive later on

System Settings

Create: Virtual Machine ⓧ

General OS **System** Disks CPU Memory Network Confirm

Graphic card: Default ▾ SCSI Controller: VirtIO SCSI single ▾

Machine: Default (i440fx) ▾ Qemu Agent: ☒

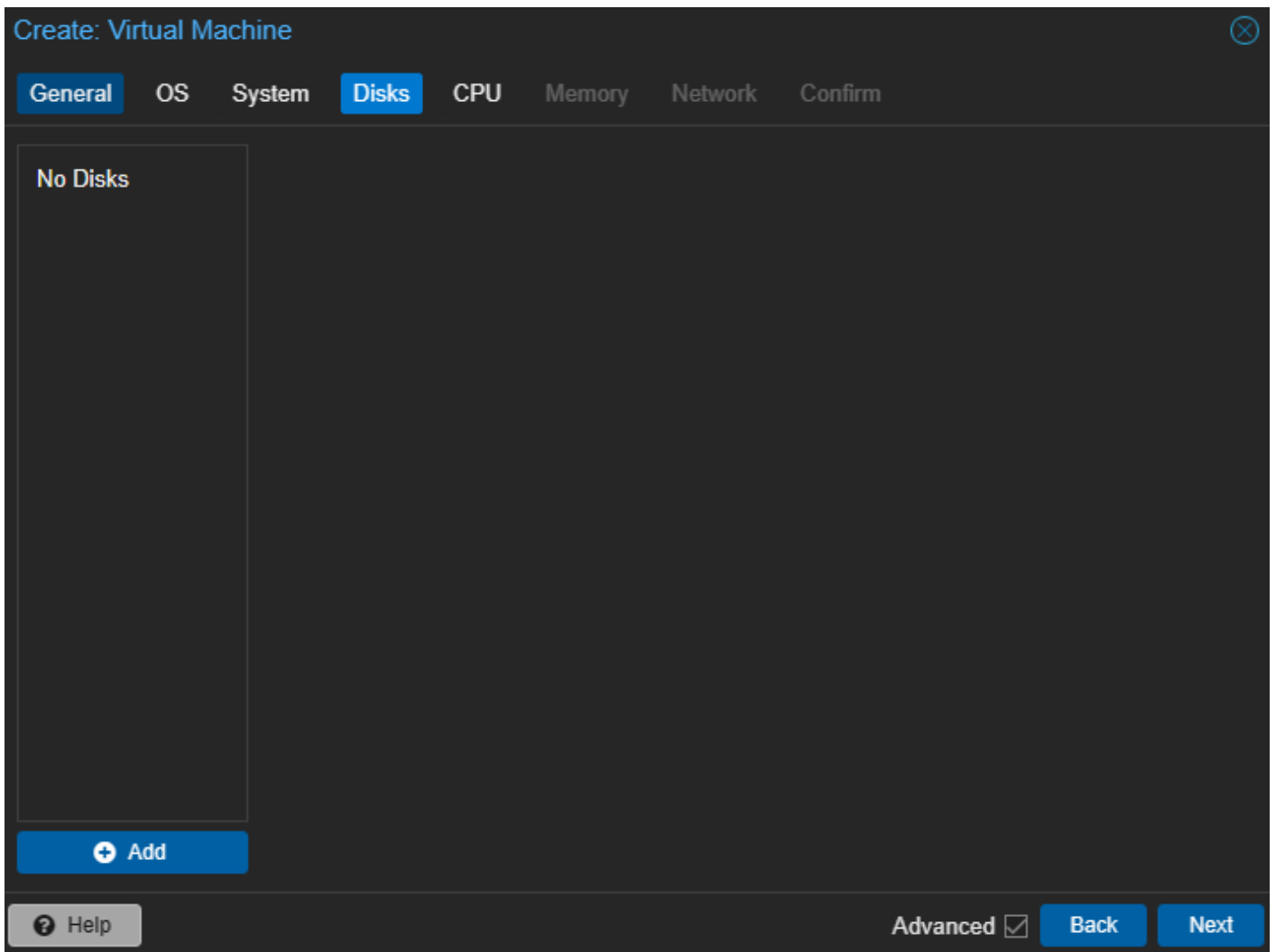
Firmware

BIOS: Default (SeaBIOS) ▾ Add TPM: ☐

ⓧ Help Advanced ☒ Back Next

- Select qemu agent
- Keep everything else as default or change if you'd like

Disk Settings



- Delete the disk, we'll add this later on

CPU Settings

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

Sockets:

1

Type:

x86-64-v2-AES

Cores:

1

Total cores:

1

VCPUs:

1

CPU units:

100

CPU limit:

unlimited

Enable NUMA:

☐

CPU Affinity:

All Cores

Extra CPU Flags:

Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	md-clear	Required to let the guest OS know if MDS is mitigated correctly
Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	pcid	Meltdown fix cost reduction on Westmere, Sandy-, and IvyBridge Intel CPUs
Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	spec-ctrl	Allows improved Spectre mitigation with Intel CPUs
Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	ssbd	Protection for "Speculative Store Bypass" for Intel models
Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	ibpb	Allows improved Spectre mitigation with AMD CPUs
Default	-	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	+	virt-ssbd	Basis for "Speculative Store Bypass" protection for AMD models

Help

Advanced ☒

Back

Next

You can keep this at default settings, or change depending on what your intentions are for this template. I plan to use this specific template for Kubernetes nodes, so I'll be using minimal requirement specs

Memory Settings

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

Memory (MiB):

2048

Minimum memory (MiB):

2048

Shares:

Default (1000)

Ballooning Device:

☒

Help

Advanced ☒

Back

Next

- I kept this default at 2 GB for memory. You can always scale up your clones: keep in mind that this is just a base configuration

Network Settings

Create: Virtual Machine

General

OS

System

Disks

CPU

Memory

Network

Confirm

☐ No network device

Bridge:

vmbr0

Model:

VirtIO (paravirtualized)

VLAN Tag:

no VLAN

MAC address:

auto

Firewall:

☒

Disconnect:

☐

Rate limit (MB/s):

unlimited

MTU:

1500 (1 = bridge MTU)

Multiqueue:

?

 Help

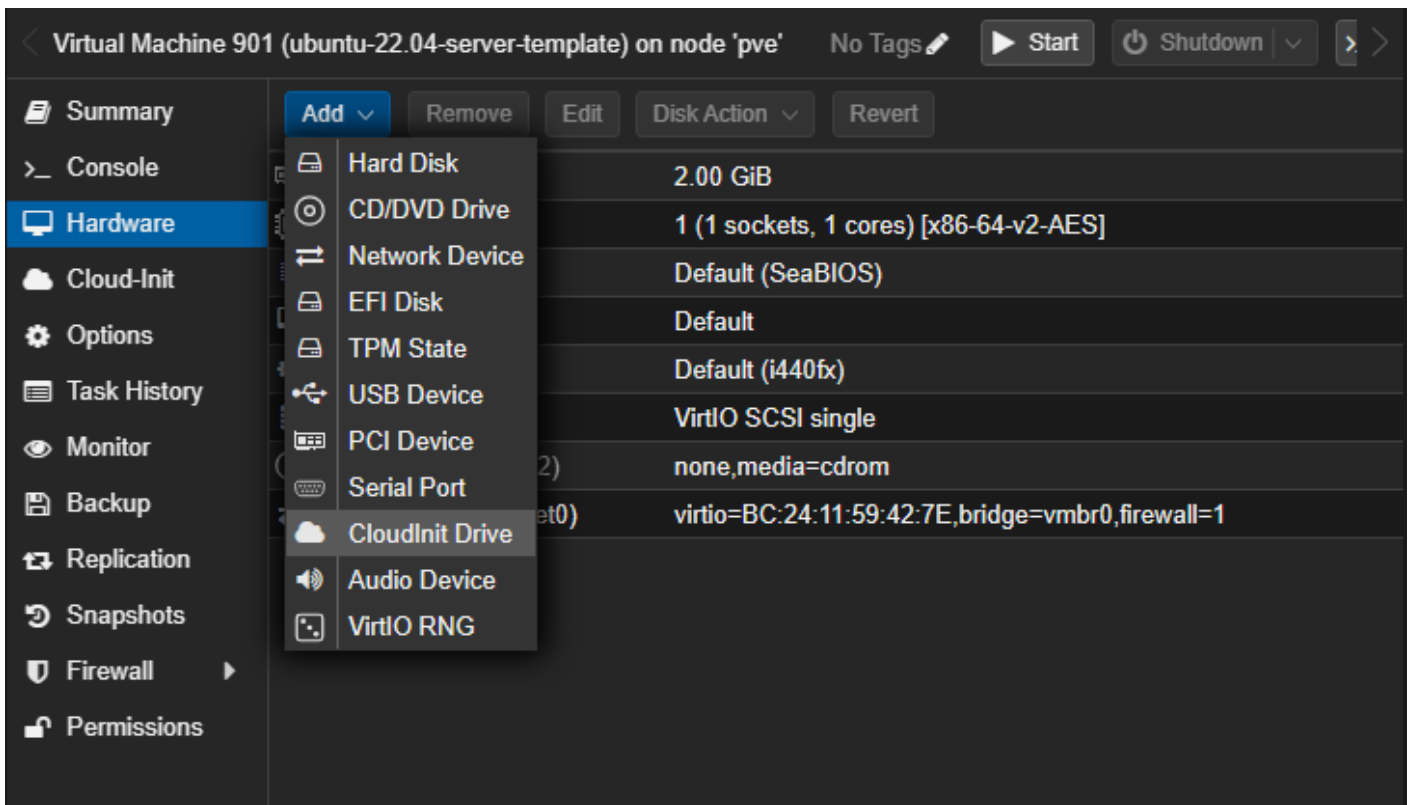
Advanced ☒

Back

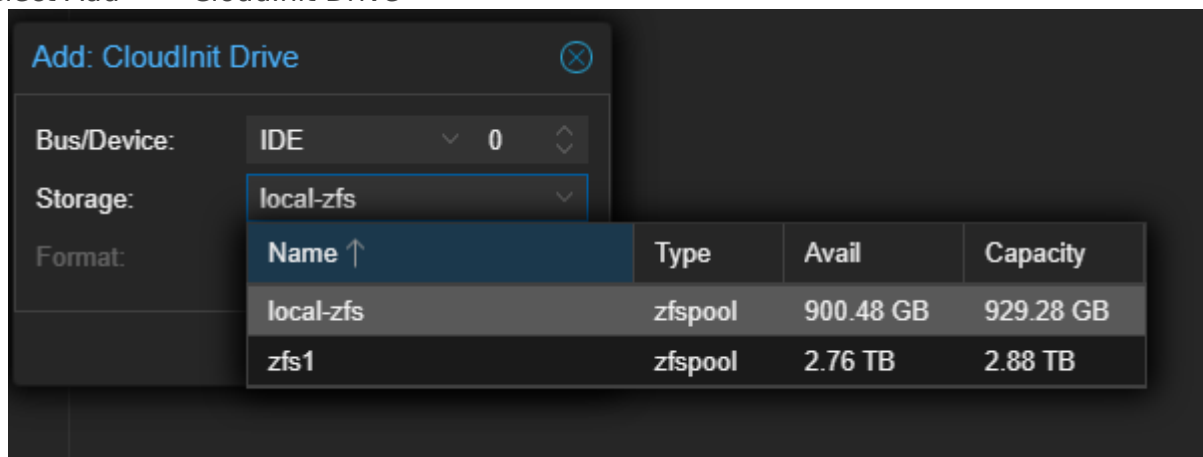
Next

- I kept my network settings the same, as I manage my homelab network with Static Leases and VLANS through OPNsense
- After this step, confirm and create your VM

Hardware Settings

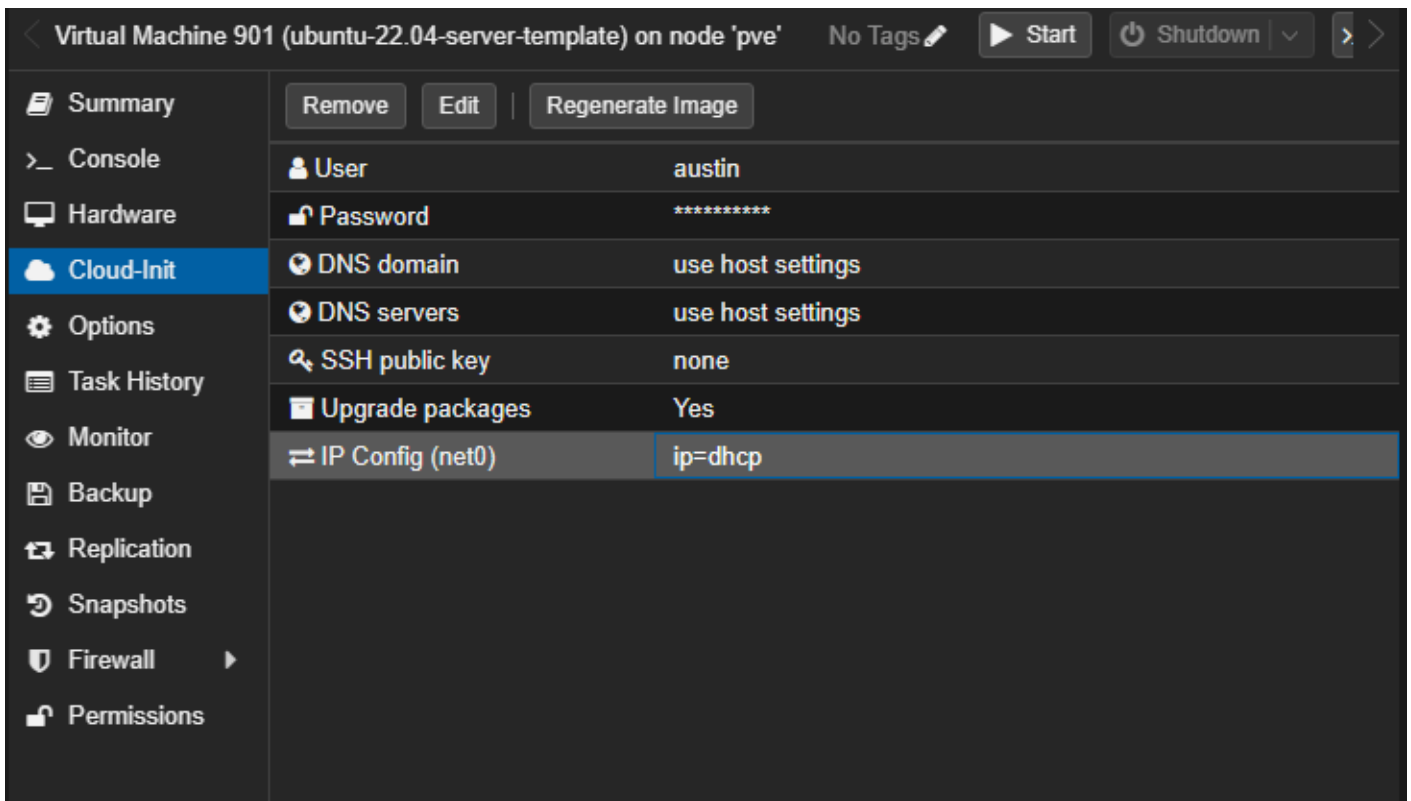


- Select your VM and navigate to the Hardware tab
- Select Add ---> CloudInit Drive



- I selected local-zfs for my storage

Cloud-Init Settings



- Navigate to Cloud-Init
- Configure settings however you'd like your VMs to be created
- I set username, password, upgrade packages, and IP to DHCP

Establish a terminal connection to your Proxmox VE so you can configure Cloud-Init further

```
Using username "root".
root@192.168.2.11's password:
Linux pve 6.5.13-3-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.13-3 (2024-03-20T10:45Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Apr  3 03:23:53 2024 from 192.168.1.198
root@pve:~#
```

You'll now want to download the cloud image you'd like to use for the VM with wget. You can find Cloud Images at the [Ubuntu Cloud website](#) and select any version of flavor of linux you'd like. I'll be using the Ubuntu Minimal 22.04 LTS (Jammy) image:

Ubuntu Minimal Cloud Images (RELEASED)

Ubuntu Minimal Cloud Images are official Ubuntu images and are pre-installed disk images that have been customized by Ubuntu engineering to have a small runtime footprint in order to increase workload density in environments where humans are not expected to log in.

For more information, please see the following:

- [Ubuntu Minimal wiki page](#)
- [Ubuntu Cloud Portal](#)
- [Commercial Support Options](#)
- [Community Help Page](#)

Cloud image specific bugs should be filed in the [cloud-images](#) project on Launchpad.net.

Launching Ubuntu Minimal

KVM

When launching the download image from KVM, you will need to specify the virtio network driver.

LXD

First add the new Ubuntu Minimal images simplestreams endpoint:














```
lxc remote add --protocol simplestreams ubuntu-minimal https://cloud-images.ubuntu.com/minimal/releases/
```

Launch the 22.04 Jammy Jellyfish minimal image:

```
lxc launch ubuntu-minimal:jammy
```

Ubuntu Minimal 22.04 LTS (Jammy Jellyfish)

Select the release you'd like to use, or scroll all the way down and select release/ for the most current. In this folder, find the amd64.image and copy the download link.

Name	Last modified	Size	Description
 Parent Directory		-	
 MD5SUMS	2024-03-20 00:43	567	
 MD5SUMS.gpg	2024-03-20 00:43	833	
 SHA256SUMS	2024-03-20 00:43	791	
 SHA256SUMS.gpg	2024-03-20 00:43	833	
 ubuntu-22.04-minimal-cloudimg-amd64-lxd.tar.xz	2024-03-20 00:43	412	Ubuntu Server mini
 ubuntu-22.04-minimal-cloudimg-amd64-root.manifest	2024-03-20 00:43	11K	Package manifest f
 ubuntu-22.04-minimal-cloudimg-amd64-root.tar.xz	2024-03-20 00:43	103M	Ubuntu Server mini
 ubuntu-22.04-minimal-cloudimg-amd64.img	2024-03-20 00:43	284M	QCow2 UEFI/GPT Boc
 ubuntu-22.04-minimal-cloudimg-amd64.manifest	2024-03-20 00:43	12K	Package manifest f
 ubuntu-22.04-minimal-cloudimg-amd64.squashfs	2024-03-20 00:43	135M	Ubuntu Server mini
 ubuntu-22.04-minimal-cloudimg-amd64.squashfs.manifest	2024-03-20 00:43	11K	Package manifest f
 unpacked/	2024-03-20 00:43	-	

In your terminal window for Proxmox, run the command below:

```
wget https://cloud-images.ubuntu.com/minimal/releases/jammy/release/ubuntu-22.04-minimal-cloudimg-amd64.img
```

After downloading, run the ls command to view the file name:

```
root@pve:~# ls
ubuntu-22.04-minimal-cloudimg-amd64.img  ubuntu-kube-node.qcow2
ubuntu2204server-template.qcow2
root@pve:~#
```

Since I've already created templates before, I have some files already in there. The file I just downloaded is ubuntu-22.04-minimal-cloudimg-amg64.img

Before moving forward, run the following command:

```
qm set 901 --serial0 socker --vga serial0
```

- This is required so that your Proxmox VM console will show your VM output
- 901 should be changed to whatever your VM ID is

Next you'll want to rename the file and change the file extension to .qcow2

Run the following command:

```
mv ubuntu-22.04-minimal-cloudimg-amg64.img ubuntu-min-2204server.qcow2
```

Set the disk size with the following command:

```
qemu-img resize ubuntu-min-2204server.qcow2 32G
```

- This will set the base hard drive size to 32Gb. You can always change this later

Next, you'll want to run the following command to import ther drive to your VM in Proxmox:

```
qm importdisk 901 ubuntu-min-2204server.qcow2 local-zfs
```

- Set 901 to your VM ID
- Set local-zfs to your storage
 - If you're not using zfs, yours is most likely local-lvm

Next, navigate in your Proxmox GUI to the Hardware section of your VM, and you'll find "unused disk 0" at the bottom. Select edit, and check the following boxes if you're using SSDs in your host machine:

SSD emulation:	<input checked="" type="checkbox"/>	Backup:	<input checked="" type="checkbox"/>
Read-only:	<input type="checkbox"/>	Skip replication:	<input type="checkbox"/>
		Async IO:	Default (io_uring) ▼

Help
Advanced ☒
OK
Reset

Cache:	Default (No cache) ▼
Discard:	<input checked="" type="checkbox"/>
IO thread:	<input checked="" type="checkbox"/>

The last step before starting the VM is to edit the Boot Order. You can do this by navigating to the Options tab within your VM.

Edit: Boot Order ✕

#	Enabled	Device	Description
≡ 1	<input checked="" type="checkbox"/>	⊙ ide2	none,media=cdrom
≡ 2	<input checked="" type="checkbox"/>	🗄 scsi0	local-zfs:base-900-disk-0,discard=on,iothread=1,size=32...
≡ 3	<input checked="" type="checkbox"/>	↔ net0	virtio=BC:24:11:C2:BB:49,bridge=vmbro,firewall=1
≡ 4	<input checked="" type="checkbox"/>	⊙ ide0	local-zfs:vm-900-cloudinit,media=cdrom

Drag and drop to reorder

Help
OK
Reset

- Edit it so that scsi0 is enabled and moved up to 2nd

With all your configurations set, right-click on the machine and select "Convert to Template". To test everything's worked correctly, right click on the template and select "Clone" then select "Full Clone" for the mode:

Target node:	pve ▼	Mode:	Full Clone ▼
VM ID:	101 ▼	Target Storage:	Same as source ▼
Name:		Format:	QEMU image format (qc ▼
Resource Pool:			

Start your machine and test that it's working as configured!

Revision #2

Created 3 April 2024 02:47:35 by Austin

Updated 5 April 2024 01:31:05 by Austin