

Migration of servers to Proxmox VE

Contents

Introduction

Physical-to-Virtual (P2V)

- Clonezilla Live CDs

- VMware Converter

 - Prepare Windows

 - Prepare location to save local image

- VMware vCenter Settings

 - Prepare the VM on Proxmox VE

 - Move the image to the Proxmox VE Server

 - Converting to qcow2

 - Final Steps

 - Alternative Methods

Virtual-to-Virtual (V2V)

- VMware

 - Exporting

 - GUI

 - CLI

 - Importing

 - Server self-migration

- HyperV

- XEN

- FreeNAS

 - Preparation in FreeNAS

 - Importing to Proxmox VE

- RHEV

- QEMU/KVM

Further information

See Also

References

Introduction

There are various ways to migrate existing servers to Proxmox VE. They can be divided into two categories:

Physical-to-Virtual (P2V)

- Migration of physical servers to Proxmox VE

Virtual-to-Virtual (V2V)

Migration of virtual machines and containers from other hypervisors to Proxmox VE

Physical-to-Virtual (P2V)

Follow these steps to do a P2V migration and turn a physical machine into a Proxmox VE virtual machine that uses Qemu and KVM.

Clonezilla Live CDs

This method is fast, reliable and OS independent as it uses live CDs.

- Get a live Linux CD like Clonezilla (<http://clonezilla.org/>).
- Prepare the source host, make sure that you have standard IDE drivers enabled (for WinXP/Win2003 use the mergeide.reg ([File:Mergeide.zip](#))), for W2k I followed this link (solution 2 worked for me): [\[1\] \(http://www.motherboard.windowsreinstall.com/problems.htm\)](http://www.motherboard.windowsreinstall.com/problems.htm)
- Now, boot the physical host with Clonezilla, go for beginner mode and select device - device and then remote, just follow the wizard
- On the Proxmox VE host, prepare a KVM guest, make sure you got a big enough IDE disk assigned (add 1 GB extra to be on the safe side) and also boot this KVM guest with the live CD and execute a shell.
- Become root and run fdisk (fdisk /dev/sda/) to make sure that sda is here, exit fdisk with 'w'. this was essential.
- Now enter all commands from the wizard on the source server tells you (configure network and request the copy process)
- After success (the wizard just copies the data, means I got a 80 GB disk but only 5 GB data on it so it was just a few minutes for the whole process on a gigabit network) just change the boot device to the hard disk and start the VM. Windows will install all needed drivers automatically, just the Intel NIC drivers for e1000 must be loaded from ISO (I got one big driver ISO from Intel containing all NIC drivers).

VMware Converter

Physical (running) Windows server to Proxmox VE (KVM) using VMware vCenter Converter Standalone Client (V5)

Tested on an HP ProLiant ML350 G5 and G6

Prepare Windows

VMware vCenter Converter Standalone Client

- Download here (https://my.vmware.com/web/vmware/info/slug/infrastructure_operations_management/vmware_vcenter_converter_standalone/5_0)(version 5.x is Free Software and also free of charge but you need to create an account and login

before download)

Mergeide.reg

- Execute mergeide.reg ([File:Mergeide.zip](#))

Prepare location to save local image

This guide is using an external USB Hard Drive. You may also save to a mapped network share.

NOTE Although the final image will be around the same size as the actual amount of data on the server, the Proxmox VE server should have enough free space to fit the total physical disk of the server unless you plan to shrink the windows disks. (http://pve.proxmox.com/wiki/Shrink_Qcow2_Disk_Files) once migrated to Proxmox VE.

VMware vCenter Settings

Launch VMware vCenter and use the following settings:

- **Source type:** Powered-on machine
- **Specify the powered-on machine:** This local machine
- **Select destination type:** VMware Workstation or other VMware virtual machine
- **Select VMware Product:** VMware Workstation 8.0.x
- **Name:** Enter the desired name for the image
- **Select a location for the virtual machine:** Browse to the USB or Network drive where the image will be saved.

The next screen shows the settings for the virtual machine.

Click on Advanced options, select the Post-conversion tab and make sure 'Install VMware Tools on the destination virtual machine' is NOT check. We do not want to install VMware tools.

Click next and Finish.

It will now convert your physical machine to a .vmdk file.

NOTE: Depending on your hardware, you may need to boot the .vmdk file using VMware Workstation or Player before moving the file to the Proxmox VE server. This allows windows to install additional drivers for the disk controller. If promoted to convert the disk to Workstation 9.x compatibility, say Yes. You won't know if you need this step until starting the Windows VM in the final step. If you get a blue screen during boot, you should try this step.

Prepare the VM on Proxmox VE

Create a new KVM virtual machine. You'll want to use similar CPU and memory as the physical system. In the Hard Disk menu, leave everything as default. We won't be using the disk created by Proxmox VE. Finish creating the VM. Make note of the VMID. For this guide, we'll be using 100 as an example.

Once the VMware converter has completed, disable all of the networks adapters on the physical server and shut down. Disabling the network adapters will avoid potential IP conflicts if you will start the physical server back into Windows after you have your new virtual server running.

Move the image to the Proxmox VE Server

Plug a USB Hard Drive into the server

From the Proxmox VE command line:

```
mkdir /mnt/usb
```

```
mount /dev/sdc1 /mnt/usb/
```

```
ls /mnt/usb
```

You should see the contents of the USB drive. In my case, the vmdk file was located in /mnt/usb/windows-server/

Converting to qcow2

We want to convert the .vmdk image file to qcow2. To do this, use the following command:

```
qemu-img convert -f vmdk /mnt/usb/windows-server/windows-server.vmdk -O qcow2 /var/lib/vz/images/100/windows-server.qcow2
```

This can take a while depending on the size of file and speed of your system.

Final Steps

Once the conversion is complete, we need to edit the configuration file for the VM.

```
nano /etc/pve/local/qemu-server/100.conf
```

In the line with ide0: we want to change vm-100-disk-1.raw,size=32G to windows-server.qcow2

You may delete the empty disk created by Proxmox VE when you created the VM.

```
rm /var/lib/vz/images/100/vm-100-disk-1.raw
```

Start the VM and open the console. Windows should boot up normally. It'll take a few minutes to detect the hardware changes. If the Windows system had a static IP address, you'll have to reconfigure the settings.

Alternative Methods

- [OpenVZ Template Creators \(https://github.com/QuinnyPig/OpenVZ-Template-Creators\)](https://github.com/QuinnyPig/OpenVZ-Template-Creators) to create non Debian (we have [DAB](#) for it) templates.

Virtual-to-Virtual (V2V)

Follow these steps to do a V2V migration and move a virtual machine from another hypervisor to a Proxmox VE virtual machine that uses Qemu and KVM.

VMware

This explains the migration from a VMware ESXi 6.7 hypervisor to Proxmox VE 6.1. It is tested with guests with the following operating systems:

- Debian 10
- Windows 10 Pro
- Windows Server 2016 Standard

Exporting

GUI

In VMware ESXi navigate to your virtual machine in the tree on the left. Choose Actions->Export. This will download disk images and information about your virtual machine in the .ovf format in the browser. Move those files to a storage that is accessible by your Proxmox VE host.

CLI

Install VMware's ovftool (<https://my.vmware.com/group/vmware/get-download?downloadGroup=OVFTOOL440>) on your Proxmox VE host. ovftool version 4.4 has been reported to work with the following versions of ESXi: 6.5 and 6.7. Others (for example, 6.0 (<https://forum.proxmox.com/threads/unable-to-migrate-vm-from-esxi-6-0-0-to-proxmox-6-3-2.82301/post-362781>)) might crash with an unhelpful error message

Remove any attached disk or ISO from your ESXi VM and run

```
ovftool vi://root@<ip-of-esxi>/<name-of-a-virtual-machine> .
```

to export a virtual machine from ESXi directly into your current directory.

You can replace the dot with any other path, for example `"/mnt/pve/<some-storage>".` This way you can export directly to a storage that you created in Proxmox VE.



Note: You have to extract .ova archives

With some configurations you get a single (.ova) archive file that contains all .vmdk disk images and the small XML description .ovf file. If this is the case, then you have to extract it before you continue with `qm importovf`.

Importing

Go to the command line interface of Proxmox VE. Use the command `qm importovf <new-vmid> </path/to/file.ovf> <storage>` to import the virtual machine. For example:

```
qm importovf 200 /tmp/exported-vm.ovf local-lvm
```

This will create a new virtual machine using cores, memory and VM name as read from the OVF manifest, and import the disks. You have to configure the network manually. You can find syntax and an example on how to use this command on its [man page \(https://pve.proxmox.com/pve-docs/qm.1.html\)](https://pve.proxmox.com/pve-docs/qm.1.html).



Note: Windows guests require a few additional steps

If you are migrating a Windows guest, you additionally have to execute the following commands. This example assumes that your imported virtual machine has the ID 130.

- Use UEFI instead of BIOS for the virtual machine:

```
qm set 130 --bios ovmf
```

- Windows requires additional drivers to use SCSI. Therefore, you should temporarily use SATA instead. In the following command, 'scsi0' should be replaced with the drive that's used to boot Windows.

```
sed -i 's/scsi0:/sata0:/' /etc/pve/qemu-server/130.conf
```

This will give you a initial working version. You can then improve your experience and access remaining SCSI drives by installing additional drivers as explained in [Windows 10 guest best practices](#).

Server self-migration

It is also possible to migrate without the need to export each VM separately including virtual disks. This way, you can convert a server from vSphere to Proxmox VE without the need of a second server. For this process your vSphere should use VMFS6 and you need at least one empty HDD.

1. Export the VM information without the disks using ovftool (you still need to configure the network configuration for each VM).

```
ovftool -NoDisks vi://root@<ip-of-esxi>/<name-of-a-virtual-machine> .
```

2. Install Proxmox VE on some disk that does not contain any important data. You don't need vSphere anymore at this point. If you have an OS disk with only vSphere on it, then you can now overwrite it with Proxmox VE.

Warning: Do not use/touch any other existing drives which are VMFS formatted

3. Create a directory on the above mention spare HDD.

4. Install vmfs6-tools which you need to mount (ready-only) the VMFS-formatted drives with the vSphere virtual disks.

```
apt install vmfs6-tools -y
```

5. List all available drives to identify the VMFS formatted ones

```
fdisk -l
```

6. Mount the VMFS disk partition (note it is read only)

```
vmfs6-tool /dev/<device><partition>
```

7. Convert the vSphere disk to a suitable format for Proxmox VE

```
qemu-img convert -f vmdk <path to your vmdk file>.vmdk -o <raw/qcow2> <path to your empty  
directory>.raw/qcow2 -p
```

8. While the conversion is in progress you may create the 1st VM from ovf

```
qm importovf 200 /tmp/exported-vm.ovf
```

9. As soon as the conversion is finished you may mount the new Proxmox VE disk image to the VM.

```
qm rescan
```

If all VM images have been moved away from a VMFS6 disk, you can format it and use it at Proxmox VE

HyperV

This explains the migration from a Hyper-V on Windows 10 hypervisor to Proxmox VE 6.1. It is tested with a Proxmox VE 6.1 guest.

Proxmox VE uses `qemu-img` (<https://manpages.debian.org/testing/qemu-utils/qemu-img.1.en.html>) to import disks. In my tests it supported `.vhdx` even though some sources (<https://linux.die.net/man/1/qemu-img>) don't explicitly list `.vhdx` as supported format. There are additional options if this should not work anymore:

- Convert-VHD (<https://docs.microsoft.com/en-us/powershell/module/hyper-v/convert-vhd?view=win10-ps>) is a command line tool that is built-in in Windows and converts `.vhdx` to `.vhd`. This format is supported by `qemu-img`.
- StarWind V2V Converter (<https://www.starwindsoftware.com/starwind-v2v-converter>) is a graphical tool that can create `.qcow2` disks from Hyper-V guests.

In Hyper-V-Manager choose "Export..." at the right side and then a destination. While many files (for example for snapshots) will be exported, we are only interested in the `.vhdx` file(s). Move those files to a storage that is accessible by your Proxmox VE host. You will have to recreate the virtual machine settings yourself.

Go to the GUI of Proxmox VE and create a new virtual machine. We don't need the hard disk that the virtual machine creation wizard created. Delete it in the hardware options of the virtual machine.

Now go to the command line interface of Proxmox VE. Use the command "`qm importdisk`" to import the `.vhdx` virtual hard disk. You can find syntax and an example on how to use this command on its man page.

XEN

XEN also uses qemu disk format, so it should work in the same manner as described under "[VMware to Proxmox VE](#)".

You could use [xenmigrate](#) to do it.

FreeNAS

Those are the necessary steps to migrate a Ubuntu Bionic VM from FreeNAS 11.2 to Proxmox VE 6.2-1. The VM in FreeNAS was created with the following parameters

- Boot Loader Type: UEFI
- Guest OS: Ubuntu Bionic
- Disk
 - Disk Mode: AHCI
 - Zvol: test/ubuntu-1xmtpt

Check the name of your zvol by going to Virtual Machines → Options of the VM → Devices → Options of your disk → Edit → Zvol

Preparation in FreeNAS

1. Create a shared directory in Sharing → Unix (NFS) Shares with path /mnt/test.
2. Enable SSH in Services & edit the SSH service (Actions) to allow password login for root
3. Copy the zvol to the shared directory

1. Log in to FreeNAS via SSH

```
ssh root@ip.of.your.freenas
```

2. Copy the zvol to the shared directory

```
dd if=/dev/zvol/test/ubuntu-1xmtpt of=/mnt/test/ubuntu.raw bs=1m
```

Importing to Proxmox VE

1. Create a virtual machine (here vmid is 103) in Proxmox VE. Make sure to set BIOS to OVMF (this is UEFI).
2. Delete the disk that was created in step 1.
3. Create a directory `mkdir /home/user/freenas`
4. Mount the shared directory from FreeNAS

```
sudo mount -t nfs 192.168.31.241:/mnt/test /home/user/freenas
```

5. Import the image of the FreeNAS VM to the Proxmox VE VM as unused disk (vmid 103, storage local)


```
qm importdisk 103 /home/user/freenas/ubuntu.raw local --format qcow2
```

6. In the GUI of Proxmox VE:

1. Go to the hardware view of your new virtual machine
2. Set the display to spice
3. Double click on the unused disk to attach it and choose Virtio as bus
4. Go to the options view of your new virtual machine
5. Choose your new virtio disk as bootdisk

RHEV

The last released version of Red Hat Enterprise Virtualization (RHEV) is out of full support since August 2022^[1] and you might want migrate to a supported Proxmox VE releases earlier than it's EOL date.

To move existing VMs you need to export them as OVA (https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/html/virtual_machine_management_guide/sect-exporting_and_importing_virtual_machines_and_templates#Exporting_a_virtual_machine_to_a_host) (archived docs (https://web.archive.org/web/20221220103326/https://access.redhat.com/documentation/en-us/red_hat_virtualization/4.3/html/virtual_machine_management_guide/sect-exporting_and_importing_virtual_machines_and_templates#Exporting_a_virtual_machine_to_a_host)). You should be able to do so by selecting the virtual machine, use the **More Actions** menu and there click on **Export to OVA** and use a directory or share that PVE either can directly access or easily moved later on.

On the Proxmox VE side you can extract the OVA, for example with `tar xf exported-vm.ova` for the more common tar format (or `unzip` if compressed as zip archive).

Then you can use `qm importovf VMID /path/to/extracted.ovf` and `qm importdisk VMID PATH/TO/DISK.img STORAGE-ID`

QEMU/KVM

1. Create an new VM on Proxmox VE
2. add the existing disk image to the new VM
3. add network, usb, pcie, ... devices as they where in the original VM
4. set the boot order and adapt other options
5. start the VM

Basic Example, Step-by-Step:

First a VM has to be created, for that you need to select an unused VMID (e.g., use `pvecmd nextid`).

```
qm create 120
```

In the next step the `some-image.img` refers to the disk image you want to import and the `some-storage` refers to the name of a target storage, as listed in `pvesm status`. Please adapt those to the respective values.

```
qm importdisk 120 some-image.img some-storage
```

Please note that `qm importdisk` adds the image as unused disk to the virtual machine, you can re-attach it to the VM via the web-interface.

You also need to mark it as a boot-disk, this can be done by using `qm set 120 --boot order=scsi0`.

```
qm set 120 --scsi0 some-storage:vm-120-disk-0
```

Basic Example: All-in-One:

Since Proxmox VE 7.2 you can use the `qm create` command to achieve all steps in one go:

```
qm create VMID --scsi0 STORAGE:0,import-from=/PATH/TO/DISK.IMG --boot order=scsi0
```

Please replace ALL-CAPS parts with the respective value. After that you can adapt the remaining basics (memory, vCPU core count, vNICs, ...) via the web-interface.

Further information

If your use case is not covered by this article you should check out the [additional ways to migrate to Proxmox VE](#) in the wiki. It gathers years of knowledge for cases which are not as common as the ones explained here, but sometimes is also rather outdated.

See Also

Reference docs for importing VMs and their disk images (https://pve.proxmox.com/pve-docs/chapter-qm.html#_importing_virtual_machines_and_disk_images).

References

1. <https://access.redhat.com/support/policy/updates/rhev>

Retrieved from "https://pve.proxmox.com/mediawiki/index.php?title=Migration_of_servers_to_Proxmox_VE&oldid=11557"

This page was last edited on 13 January 2023, at 07:50.