Installation Proxmox Server 10

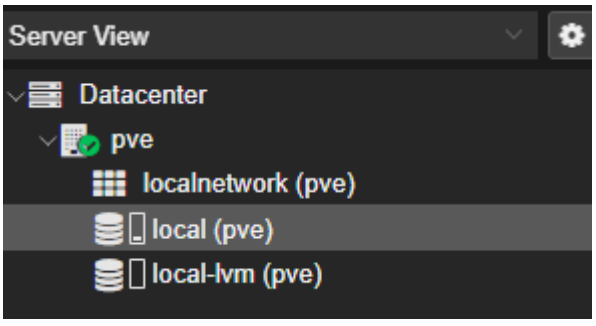
Connect to ILO 5 using 192.168.0.10, username student, password $Bt6RnWH!tgKGRJ1T!ntTf.
Click on console.
Choose HTML5.
Click on disk -> local -> choose ISO.
You choose Install Proxmox VE (Graphical)

Choose the target disk.

Set your country and time zone.
Set a password and your email
Set Hostname
Set IP-Address, Gateway and DNS Server IP
Restart the server and type the IP Address in your Browser (https://10.0.10.101:8006/)

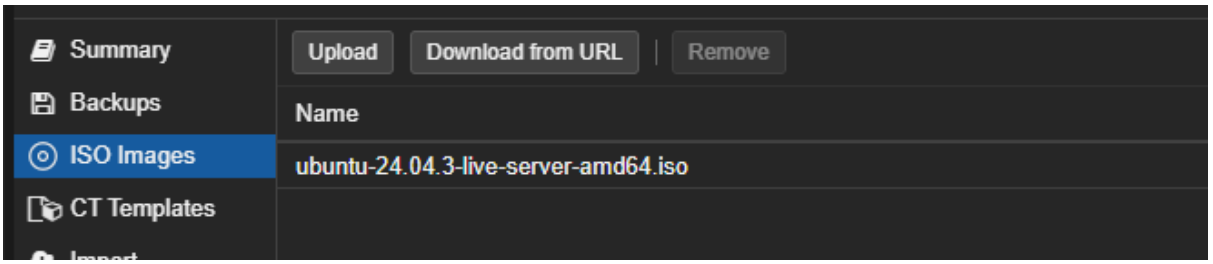
Create VM GUI

Connect to your Proxmox GUI
Under Datacenter -> Pve -> local (pve), click on ISO Images and Upload.

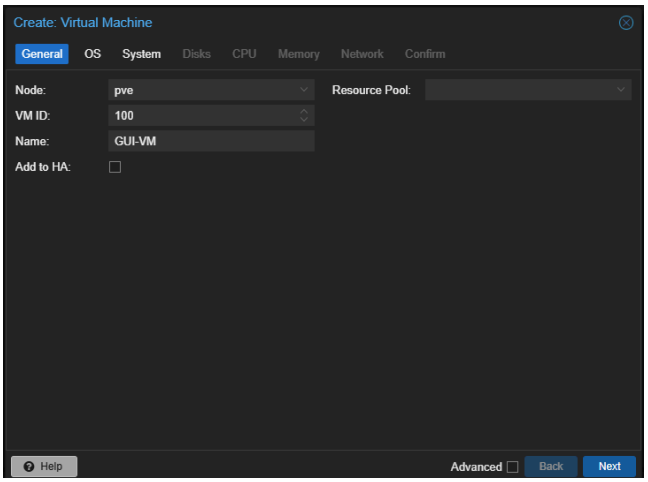


Choose the ISO file you want in the Pop up window and click upload again.

You should now be able to see your ISO file.



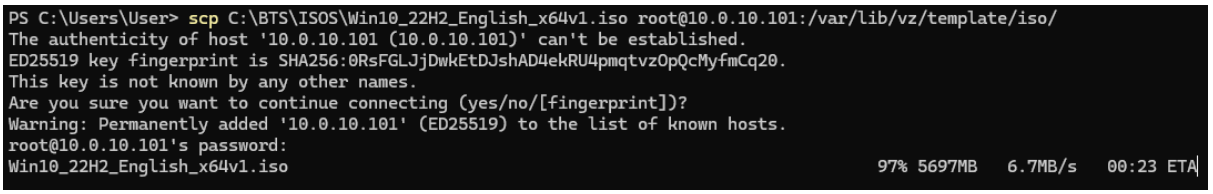
In the top right, click on Create VM.
In general, give the VM a name and leave the rest as default settings.



In OS check Use CD/DVD dis image file (iso)
In Storage choose local and in ISO image the ISO you just uploaded.
Leave the rest default and in the Confirm Tab, click finish.

Create VM CLI

If you have the ISO on your PC you need to upload it to proxmox.
Open a windows powershell and type in the command
scp C:\Users\YourName\Downloads\YourISO root@YOUR_PROXMOX_IP:/var/lib/vz/template/iso/



Connect to ILO5 and open the HTML5 terminal.
Verify the ISO is there using
ls /var/lib/vz/template/iso/

Create the VM using

*qm create 300 *
* --name "ubuntu-server" *
* --memory 2048 *
* --cores 2 *
* --net0 virtio,bridge=vmbr0 *
* --ostype l26*
You can configure your VM how you want.
Now you need to add the disks and ISO to the VM, for that use
qm set 300 --scsi0 local-lvm:20,cache=writeback
*qm set 300 --ide2 local:iso/**YourISO**,media=cdrom*
qm set 300 --boot order=ide2;scsi0
After that you can start your VM using *qm start 300*
To open access the VM terminal using the CLI you first need to add a serial port.
qm set 300 --serial0 socket --vga serial0
Now you can start your VM using
qm start 300
And acces the terminal using
qm terminal 300

Installation of a single Linux Server OS within VM on Proxmox

After creating your VM click on it on the left side and choose console.

For Windows you need the VirtIO Driver if you want it to recognize SCSI disks.
Now just proceed with your OS installation like normally.
After finishing the installation, the VM should work with the Linux Server immediately.

Installation of a single Linux Server OS within VM on Proxmox

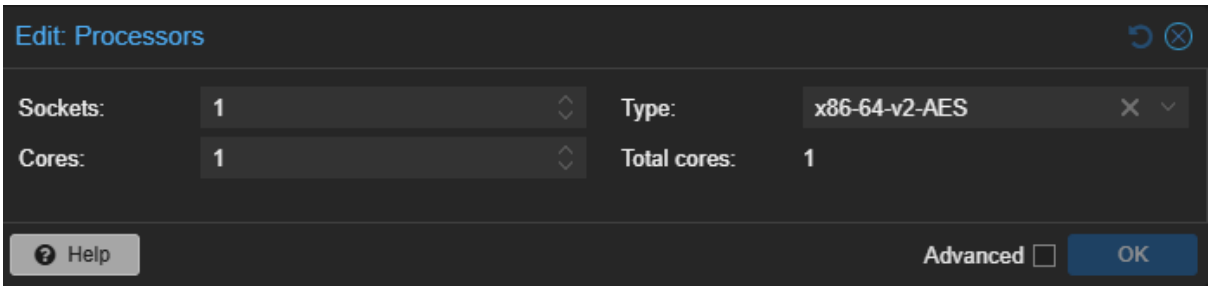
After creating your VM click on it on the left side and choose console.

Now just proceed with your OS installation like normally.
After finishing the installation, the VM should work with the Linux Server immediately.

Change the configuration of a VM in Proxmox

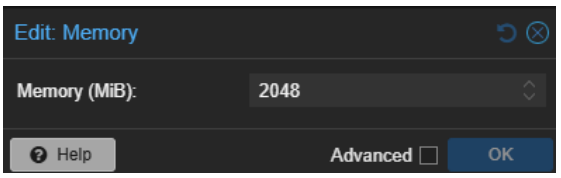
CPU

Go to your VM -> Hardware
Click Processors -> edit
Now you can change Sockets and cores.



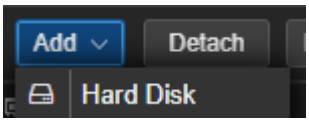
RAM

Go to your VM -> Hardware
Click Memory -> Edit
There you can change the memory size.

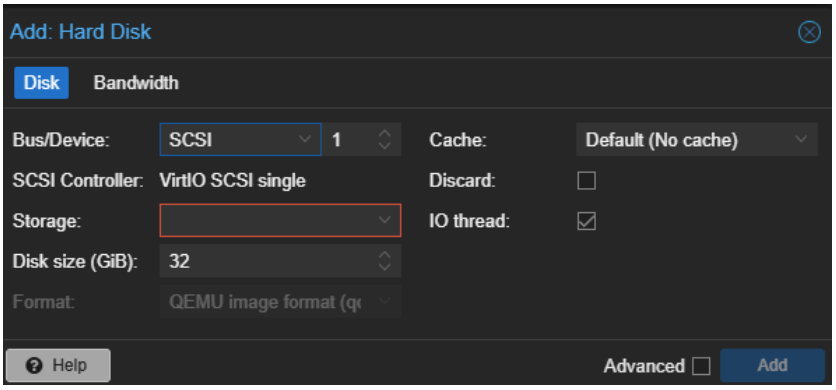


Hard Disk

Go to your VM -> Hardware
Click Add -> Hard disk



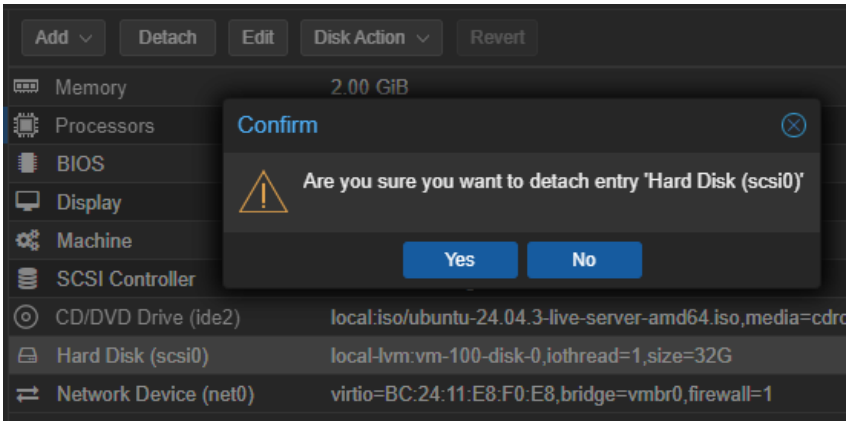
Set storage, size and bus type



Click Add

Remove Hard Disk

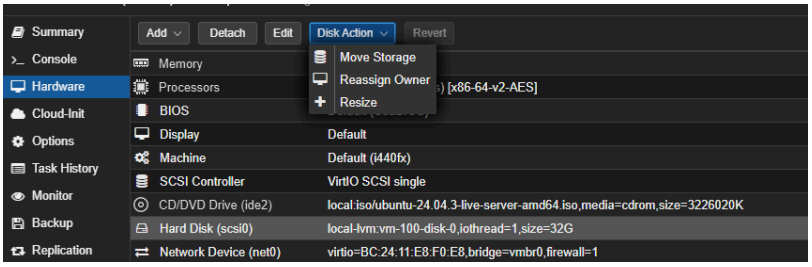
Go to Hardware
Select the disk
Click Detach -> confirm



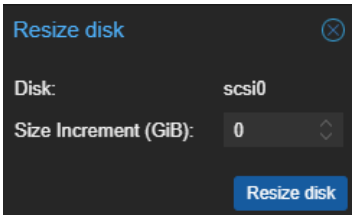
Select the detached Disk
Click Remove -> confirm

Resize Hard Disk

Go to Hardware
Select Disk
Click Disk Action -> Resize



Enter how much to add



Click Resize Disk

Duplicate a VM in Proxmox

Pick an existing VM in Proxmox and Right Click on it
Click on “Clone”
In the popup menu enter the name of the new VM and press “Clone”

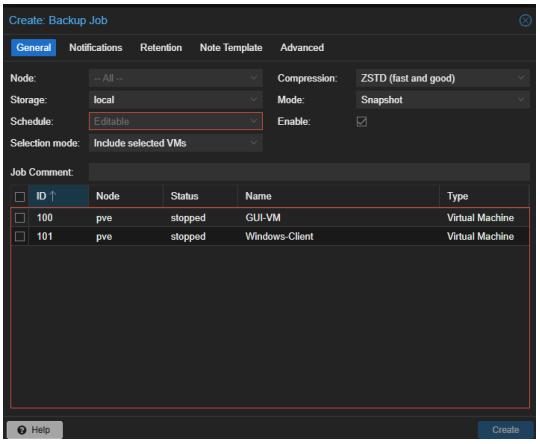
Backup the Proxmox Hypervisor configuration file

Log into to your ILO5 and open the HTML5 terminal.
Type in the command
tar czf proxmox-config-backup.tar.gz /etc/pve/ /etc/network/ /etc/hosts /etc/hostname
After this command the configuration file is backed up as ***proxmox-config-backup.tar.gz***

Backup a Proxmox VM

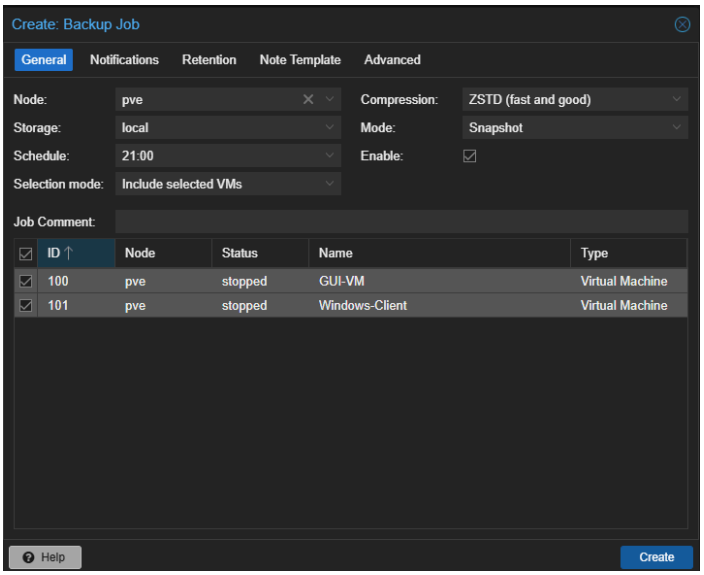
Normal VM Backup

Go to Datacenter -> Backup
Click Add



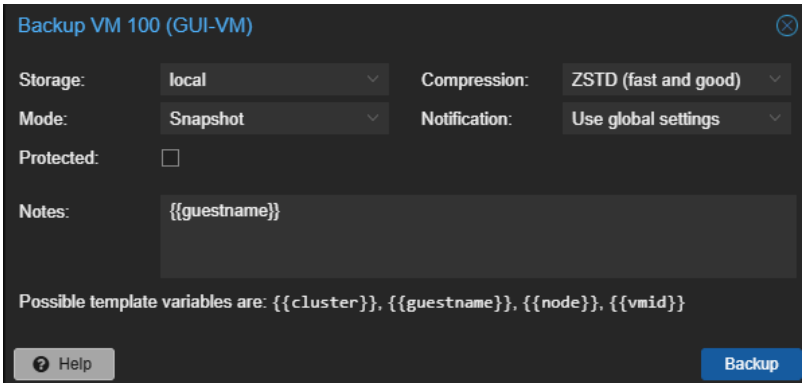
Node : What Server to backup
Storage: where to save the backup
Schedule: how often to back up
Mode: `Snapshot` (VM stays running) or `Stop` (more reliable)
Selection: which VMs to include

After everything is filled in, click create.



Immediate VM Backup

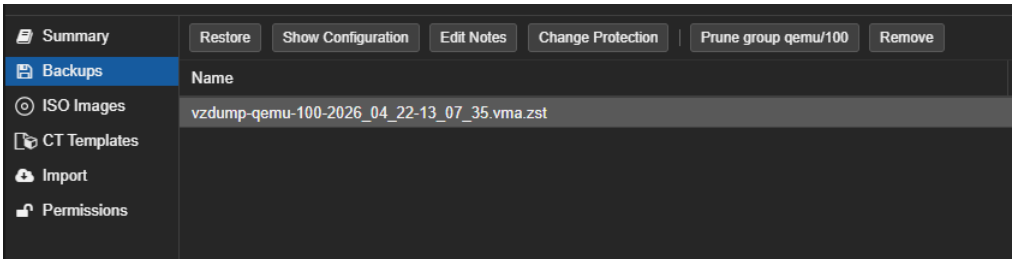
Go to your VM -> Backup
Click Backup now



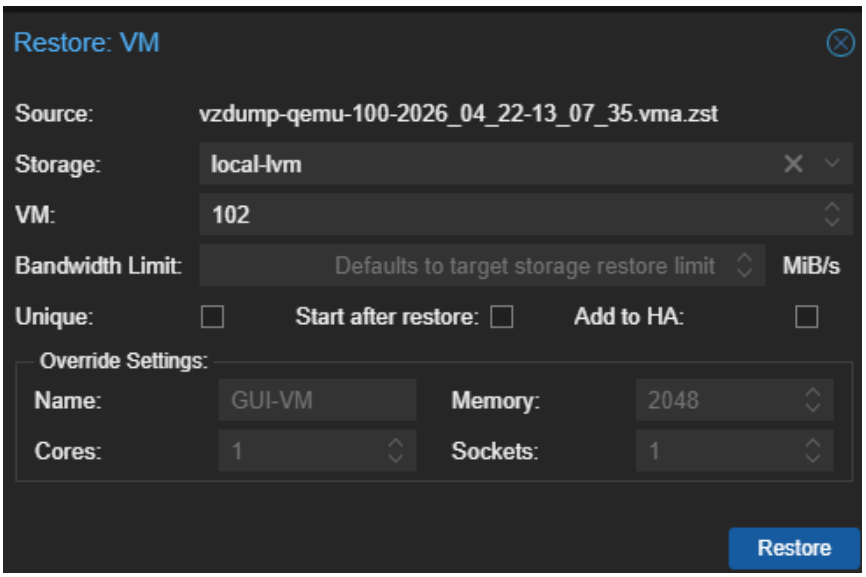
Choose storage and mode
Click Backup

Restore a VM from Backup

Go to local (pve) -> Backups
Select the backup file



Click Restore
Choose the target VM ID and Storage

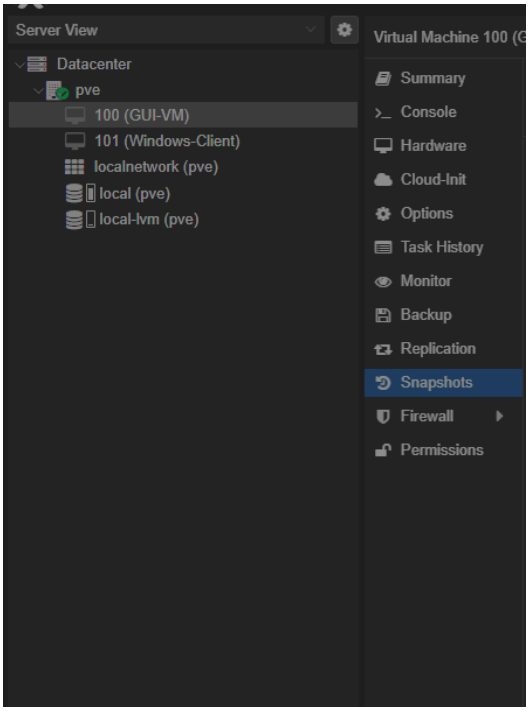


Click Restore

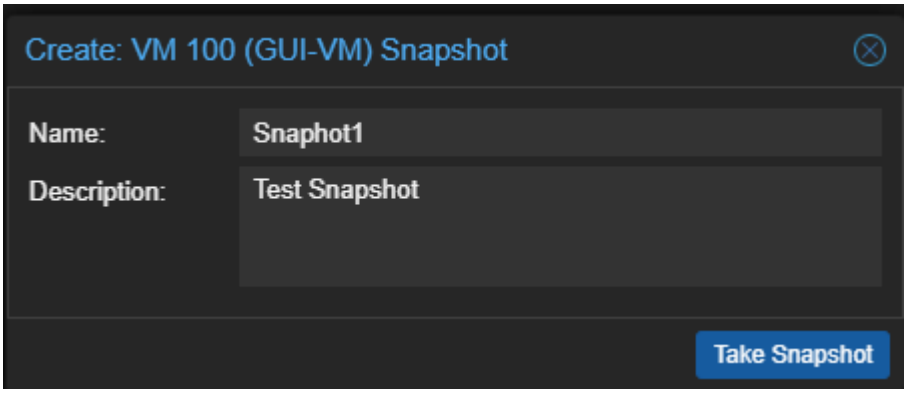
Use Snapshot of VM in Proxmox

Create Snapshot

Go to your VM
Click Snapshots



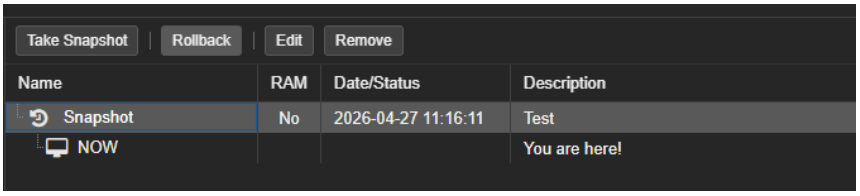
Click Take Snapshot
Give it a name and optionally a description



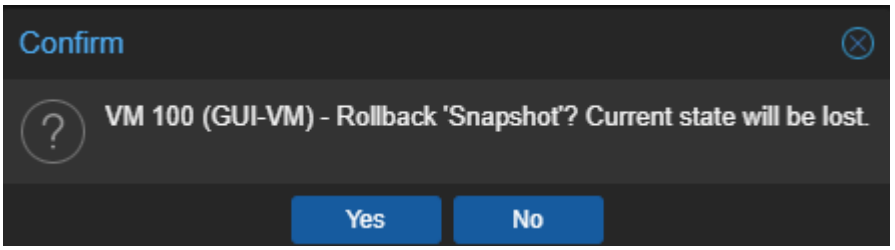
Click **Take Snapshot**

Use the Snapshot

Go to your VM
Click Snapshots
Select the snapshot you want to restore
Click Rollback

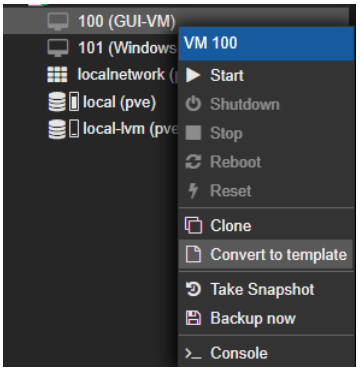


Confirm with yes



Create a VM Template in Proxmox

Choose the VM you want to use as a template and make sure it is shut down.
Right click the VM in the sidebar.
Click convert to template.



Confirm with yes

Live Migration of an VM from one Storage to another in Proxmox

Go to your truenas server.
Go to shares.
At UNIX (NFS) Shares click add.
Choose the path where you want to later save your VM.
Enable NFS.
Click on the 3 points next to your path and click edit.
At Network add the network of your Proxmox server.
Go back to your proxmox machine.
Go to datacenter
Click on add -> NFS
Give it an ID
At Server type in the IP of your truenas server.
At Export enter tha path you chose in truenas.
Click add.
Go to your VM
Click Hardware
Select the Hard disk
Click Disk Action -> Move Storage
Select the truenas storage
Check **Delete source** if you want to remove the original after moving
Click **Move Disk**
Now you moved the VM from one storage to another.

Live migration of an VM from one hypervisor to another in Proxmox

Add a cluster Network

Got to systems -> Network -> Create
Choose a linux bridge
Give it a static ip and a Gateway.
Click create

Create Cluster in Proxmox

Go to Datacenter -> Cluster
Click create cluster
Give it a name

Add a Cluster Network
Click create
Click Join Information
Click copy information
Go to your second proxmox server
Go to Datacenter -> Cluster
Click join Cluster
Paste the join information
Enter the password of server 1
Click join.

Migrate the VM

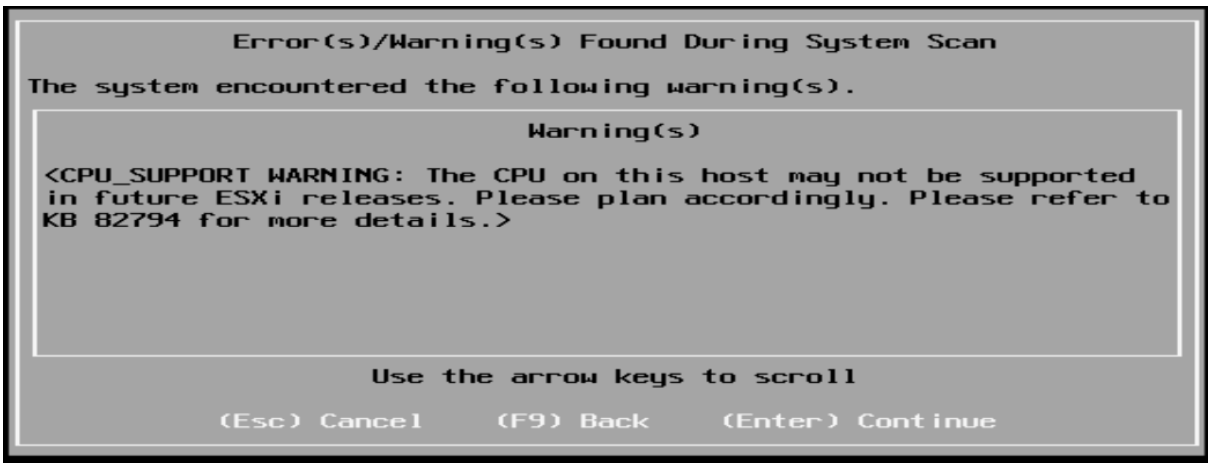
Go to your VM -> Hardware.
Remove any CD/DVD if still there.
Right click the VM.
Click Migrate
Choose a target node.
Click Migrate.

High availability Proxmox

Installation VmWare

Ilo verbinden
console anklicken
HTML 5 auswählen
Auf disk klicken → local → iso auswählen
(Da ich probleme mit dem Keyboard hatte hab eich mir eine Console Ilo App installiert und mic mit dem Remote verbunden)

Bei der installation mit enter bestätigen
F11 zum Akzeptieren der geschäftsbedinghungen
SSD Auswählen und enter
Keyboard Swiss French
Root password eingeben
Warning that CPU ist may not be supèported in future ESXI click next



Confirm install → F11
Remove installation media an reboot → enter

nach der installation wird eine IP adresse angezeigt in meinem fall https://10.0.96.155/

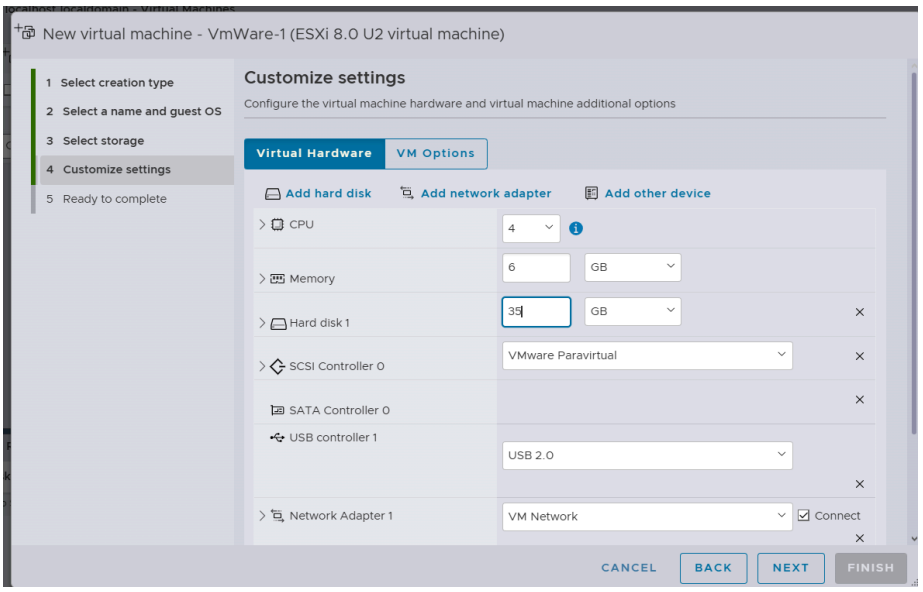
Network setting

F2 auf der console
Configure Managment Network
Ipv4 Configuration
Set static Ipv4 address
change ipv4 → enter
restart server → jetzt kann man sich mit der neuen ip adresse verbinden

Ohne vSphere Client

Vm erstellen Gui

Connect to the web Gui
Virtual Maschines → Create /register Vm
Create a new machine → Select a Name, an Os Familiy and a Os Version → select storage device → Select Hardware Power



ON the Review Tab klick Finish
Now we see the Virtual Maschine → click on it -> jetzt können wir diese verwalten

Vm erstellen CLI

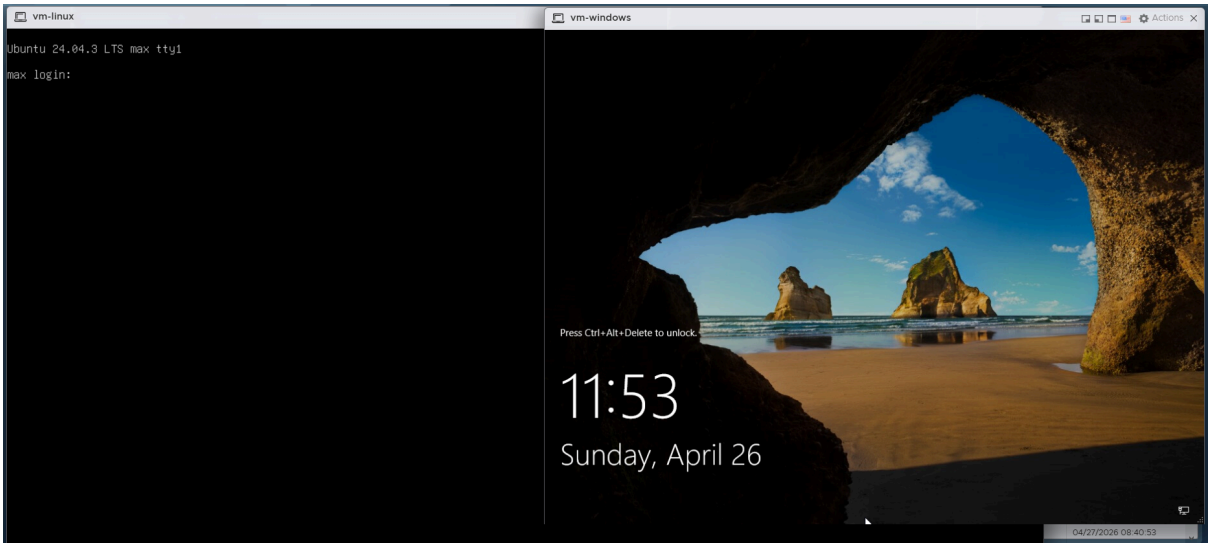
SSH Starten
Im vSphere Web Client: **Host → Manage → Services → TSM-SSH → Start**

```
# Verzeichnis anlegen
 mkdir /vmfs/volumes/datastore1/MeineVM
# Virtuelle Festplatte erstellen (z.B. 20 GB)
vmkfstools -c 20G -d thin /vmfs/volumes/datastore1/MeineVM/MeineVM.vmdk
# .vmx-Datei erstellen (Minimalbeispiel)
cat >> /vmfs/volumes/datastore1/MeineVM/MeineVM.vmx
<<EOF pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8" EOF
# VM registrieren
vim-cmd solo/registervm /vmfs/volumes/datastore1/MeineVM/MeineVM.vmx
# VM starten
vim-cmd vmsvc/power.on <vmid>
```

Add Iso

Storage → right klick datastorage 1 → Browse → Upload → select Iso Image

Windows Linux installation

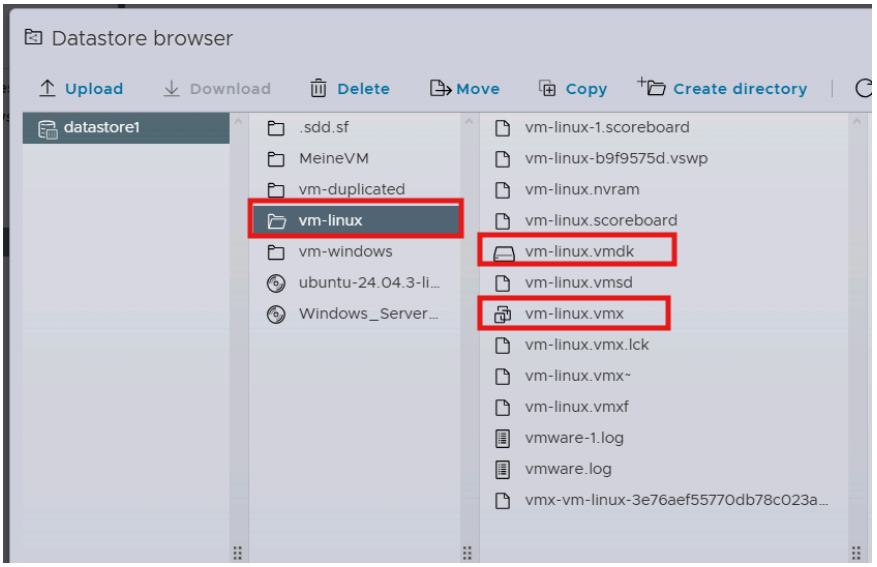


Change Hardware

Power off → Edit → change value of Resources → click Save → Start VM

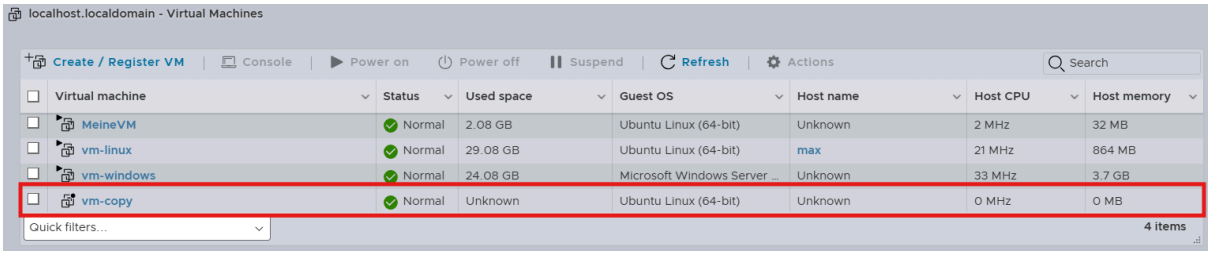
Duplicate VM

Storage → Datastore Browser → Select the Vm Directory → download .vmx and .vmdk file



Open local the .vmx
Changes:
New DisplayName
`displayName = "vm-copy"`
Need to show on the new .vmdk file
`scsi0:0.fileName = "vm-copy.vmdk"`
Need to be deleted
`uuid.bios = "..."`
`uuid.location = "..."`
`ethernet0.generatedAddress = "..."`
`ethernet0.generatedAddressOffset = "..."`

Save the file an give a new name
In DataBrowser create a new Directory for the VM and Upload the .vmdk and .vmx file
Rightklick on the .vmx and klick register the vm
Now you can see the Vm unter Virtual Maschines



Export VM

Storage → Datastore Browser → Select the Vm Directory → download

Backup the hypervisor configuration

SSH
vim-cmd hostsvc/firmware/sync_config
vim-cmd hostsvc/firmware/backup_config
Now you became a link, change the * with your ip addres
example:
http://10.0.10.111/downloads/525be5e5-333a-aa1f-c1b5-4c1ac1723ce4/configBundle-localhost.tgz

Wiederherstellen

Host in Wartungsmodus setzen
vim-cmd hostsvc/maintenance_mode_enter

Backup wiederherstellen
vim-cmd hostsvc/firmware/restore_config /tmp/configBundle-localhost.tgz

Take a Snapshot

Go to the VM where you want to take a snapshot
on The Nav bar select Actions → snapshot → Take snapshot
Give him a name and when you want a Description
Click Take Snapshot

Restore Snapshot

Go to the VM where you want to take a snapshot
on The Nav bar select Actions → snapshot → Restore snapshot
Click restore

Backup VM

Go to the VM where you want to take a snapshot
on The Nav bar select Actions → Export
Enable everything
Click Export

Restore Backup OVP

On the Sidebar click Host
On the Navbar Select Create/Register VM
Select `Deploy a virtual machine from an OVP or OVA file → Next`
Enter a name for the new vm and drag/drop the Files .vmdk and .ovf → Next

Select Where the Vm get Stored → next
Network Configuration → next
Click Finish

Vm Template via CLI

Connect via SSH to the hypervisor
Create Template Folder → mkdir /vmfs/volumes/datastore1/mein-template
Clone → vmkfstools -i /vmfs/volumes/datastore1/vm-linux/vm-linux.vmdk /vmfs/volumes/datastore1/mein-template/mein-template.vmdk -d thin

- **Source:** `/vmfs/volumes/datastore1/vm-linux/vm-linux.vmdk` – the original VM disk
- **Destination:** `/vmfs/volumes/datastore1/mein-template/mein-template.vmdk` – the cloned disk, name is freely chosen but should match the folder name for consistency
- **`-i`** – clone/import a VMDK
- **`-d thin`** – use thin provisioning (only uses actual disk space needed)

You can replace `mein-template` with any name (e.g. `ubuntu-base`, `linux-template-v1`) – just keep it consistent across all steps.

Copy VMX file → cp /vmfs/volumes/datastore1/vm-linux/vm-linux.vmx /vmfs/volumes/datastore1/mein-template/mein-template.vmx
Configure VMX file →

```
sed -i 's/vm-linux/mein-template/g' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/uuid.bios/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/uuid.location/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/vc.uuid/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/vmci0.id/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/ethernet0.generatedAddress/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/ethernet0.generatedAddressOffset/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/scsi0.sasWWID/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/sched.swap.derivedName/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/migrate.hostLog/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/extendedConfigFile/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/scsi0:0.redo/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i '/checkpoint.vmState/d' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
sed -i 's/mein-template-000002.vmdk/mein-template.vmdk/' /vmfs/volumes/datastore1/mein-template/mein-template.vmx
```

Register VM → vim-cmd solo/registervm /vmfs/volumes/datastore1/mein-template/mein-template.vmx

Create VM

1. In vSphere Client: Create/Register VM

2. Select Register an existing virtual machine or copy the template files to a new folder and register the copy
3. On first boot, select "I copied it" so ESXi generates new MAC addresses and UUIDs

Notes

- The source VM must be shut down before cloning
- When starting a copied VM, always choose "I copied it" to avoid MAC address and UUID conflicts with the original
- Do not start the template itself – always create copies and use those instead
- Since ESXi Free does not offer a native template feature, this is the recommended workaround

installation vSphere Client

Stage 1 – Appliance Deployment

1. VCSA ISO lokal mounten → installer.exe starten → **Install** klicken
2. ESXi Host: `10.0.10.121`, Port `443`, User `root`, PW `test..123` → Zertifikat bestätigen
3. VM Name: `vCenter`, Root PW: `Test..123`
4. Deployment Size: **Tiny**, Thin Disk Mode aktivieren
5. Datastore auswählen
6. Netzwerk: Static IP, Mask `255.255.0.0`, Gateway `10.0.0.1`, DNS `10.0.0.1`

Stage 2 – Konfiguration

1. NTP: `pool.ntp.org`
2. SSO Domain: `vsphere.local`, User: `administrator`, PW: `Test..123`
3. CEIP deaktivieren → **Finish** → ~15-20 Min warten

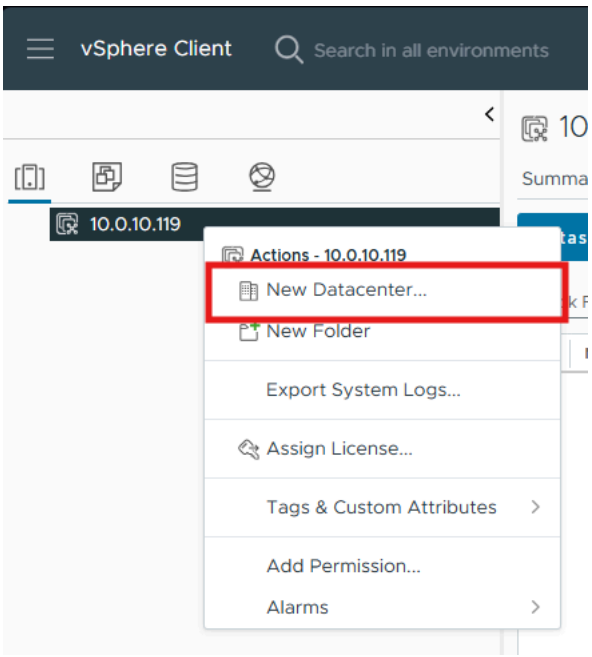
Zugang danach

https://10.0.10.212/ui` → `administrator@vsphere.local` / `Test..123`

Connect Datacenter (vSphere Client)

Connect to your ip from vSphere Client
https://10.0.10.119/ui
User: `administrator@vsphere.local`
Passwort: BTSTest..123

Right Click the Ip address and select New Datacenter…



Give them a name and click OK

Connect ESXi-Host (vSphere Client)

Rightclick on your Datacenter and select add Host
on the Wizard:

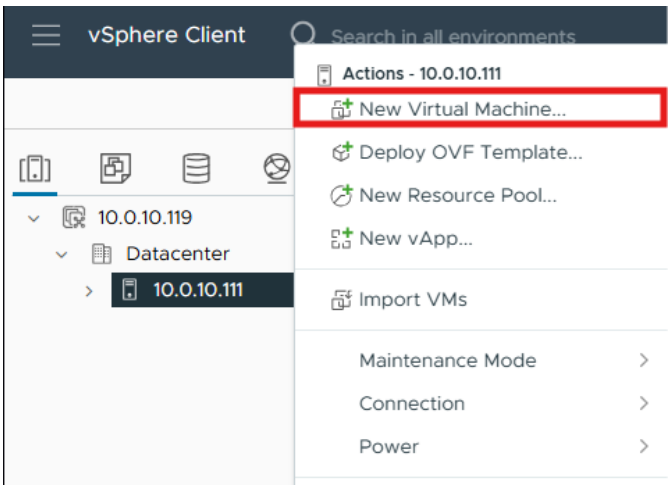
1. **Name and location:** IP oder FQDN deines ESXi-Hosts (z.B. `10.0.10.111`) → Next
2. **Connection settings:** User `root` + ESXi-root-Passwort → Next
3. **Security Alert:** Zertifikat akzeptieren → **Yes**
4. **Host summary:** kurz prüfen → Next
5. **Assign license:** Eval-Lizenz lassen → Next
6. **Lockdown mode:** *Disabled* → Next
7. **VM location:** Datacenter direkt auswählen → Next
8. **Ready to complete** → **Finish**

Upload Iso Image (vSphere Client)

Click on the arrow on Datacenter → datastore1
Click on Upload Files → select the iso Image

Create VM GUI (vSphere Client)

In the Nav Bar go to the first Symbol Hosts and Clusters
rightclick the IP address from the ESXI Host
Select New Virtual Machine



In the Wizard:

1. Select a creation type : „Create a new virtual machine" → Next
2. Select a name and folder :

- Name: ubuntu-server-01 (oder wie du willst)
- Location: dein Datacenter Lab-DC ist vorausgewählt → Next

3. Select a compute resource: Dein ESXi-Host ist vorausgewählt → Next
4. Select storage: Select Datastore → Next
5. Select compatibility: Default lassen („ESXi 8.0 U2 and later" o.ä.) → Next
6. Select a guest OS

- Guest OS Family: Linux
- Guest OS Version: Ubuntu Linux 2 (64-bit)

Next

7. Customize hardware ← das ist der wichtige Teil

Hier konfigurierst du die VM. Stell folgende Werte ein:

CPU: 2
Memory: 4096 MB (= 4 GB)
New Hard disk: 40 GB

Klick auf das Dreieck/Pfeil neben „New Hard disk", um die Details auszuklappen
Disk Provisioning: Thin Provision auswählen (spart Platz)

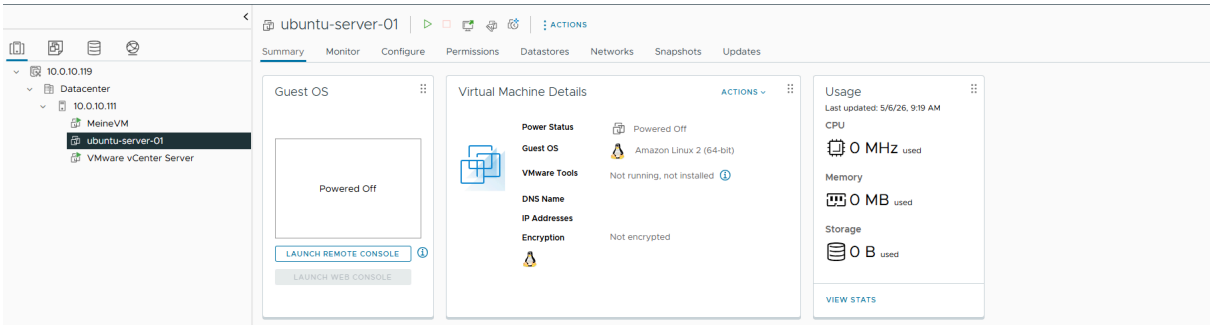
New CD/DVD Drive: ← wichtig, hier ISO einhängen!

Klick auf das Dropdown rechts → wähl „Datastore ISO File"
Es öffnet sich ein Browser → navigier zu deinem Datastore → Ordner ISOs → wähl die Ubuntu-ISO → OK

Setze ein Häkchen bei „Connect At Power On" ← unbedingt nicht vergessen, sonst bootet die VM nicht von der ISO

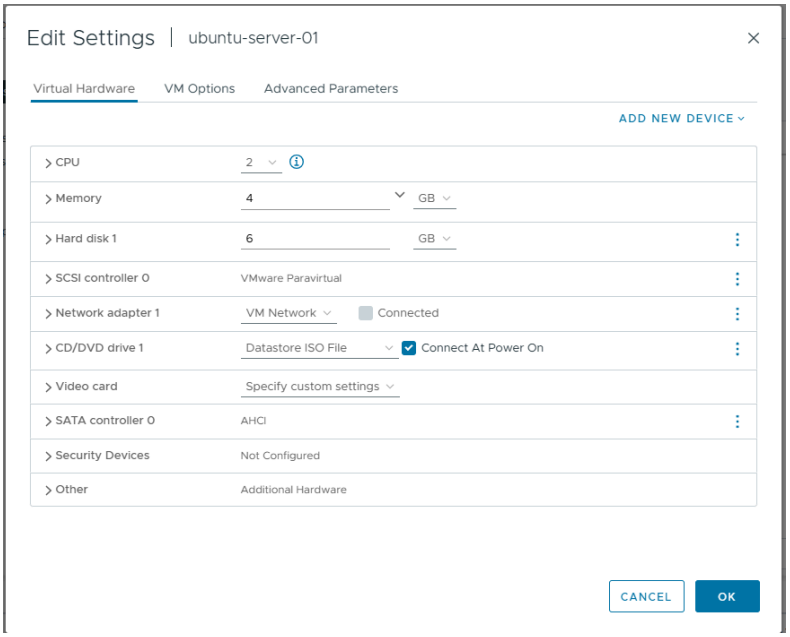
→ Next
8. Ready to complete

Übersicht aller Einstellungen prüfen
Finish



Change Hardware from a VM (vSphere Client)

Shutdown the VM: Right click on the vm → Power → Power off
Right click again on the vm → Edit Settings
Here you can change the hardware, after the changes click ok



It is possible to change the HArdware without Shutdown the vm but only Cpu and Ram

Vm Clone (vSphere Client)

Right click on ubuntu-server-01 → Clone → Clone to Virtual Machine…
In the Wizard:
1. Select a name and folder

- Name: z.B. ubuntu-server-01-clone

- Location: your Datacenter

Next

2. Select a compute resource

Dein ESXi-Host → Next

3. Select storage

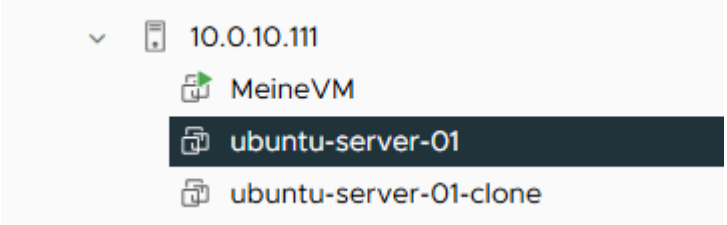
Dein Datastore → Next
(Falls du mehrere Datastores hast, kannst du den Klon auch auf einen anderen legen – nicht zwingend nötig)

4. Select clone options
Hier sind drei Häkchen, die du dir kurz anschauen solltest:

Customize the operating system: Aus. Würde Hostname/IP/SID neu setzen – ist aufwändig, weil du dafür eine „Customization Specification" anlegen müsstest. - nicht nötig

Customize this virtual machine's hardware: Aus. Lässt dich die Hardware vor dem Klonen anpassen – nicht nötig.
Power on virtual machine after creation: Aus lassen für den ersten Test.

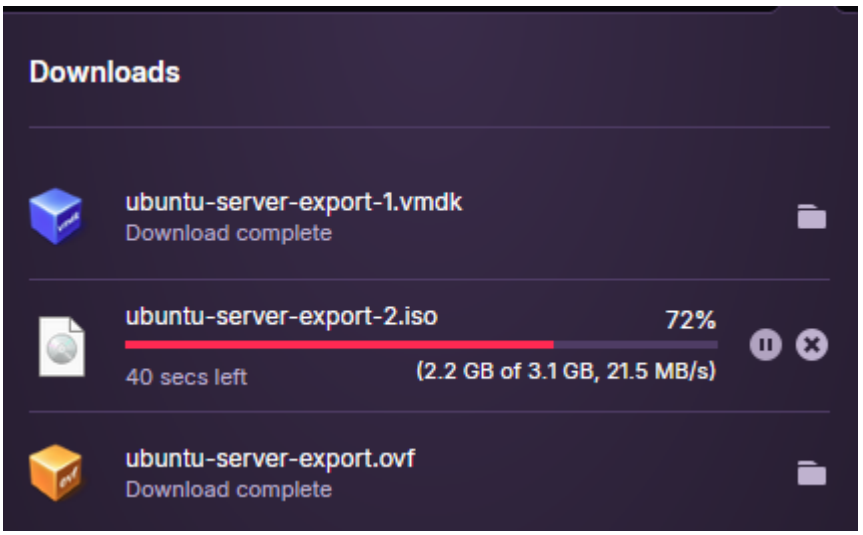
Next.
5. Ready to complete
Übersicht prüfen → Finish.



VM Exportieren (vSphere Client)

Right click on the VM → **Template** → **Export OVF Template…**
On the Popup select a name → ok

Tip: you need to allow popups



Backup vm Via Script

Connect you per ssh to the VmWare root
go to this folder: cd /vmfs/volumes/datastore1/
create a new folder: mkdir ghettoVCB
go in this folder: cd ghettoVBC

On your Device go to this Page and download the code, the file must be an .sh file
https://raw.githubusercontent.com/lamw/ghettoVCB/master/ghettoVCB.sh

When your Using MobaXterm you can put the file via drag and drop into the folder

The script need execute permissions : chmod +x ghettoVCB.sh

Now we create a folder where the backups are stored in our example is it on the same Disk/Datastore normally you use an other disk.

mkdir /vmfs/volumes/69e5cb3f-c2b3ee88-e01d-d4f5ef8b6f14/Backups

Config datei erstellen für die Backups
vi ghettoVCB.conf

Paste this text in the conf

VCB.conf << 'EOF'
VM_BACKUP_VOLUME=/vmfs/volumes/69e5cb3f-c2b3ee88-e01d-d4f5ef8b6f14/Backups
DISK_BACKUP_FORMAT=thin
VM_BACKUP_ROTATION_COUNT=3
POWER_VM_DOWN_BEFORE_BACKUP=0
ENABLE_HARD_POWER_OFF=0
ITER_TO_WAIT_SHUTDOWN=3
POWER_DOWN_TIMEOUT=5

ENABLE_COMPRESSION=0
VM_SNAPSHOT_MEMORY=0
VM_SNAPSHOT_QUIESCE=0
ENABLE_NON_PERSISTENT_NFS=0
UNMOUNT_NFS=0
EMA
IL_LOG=0
WORKDIR_DEBUG=0
VM_SHUTDOWN_ORDER=
VM_STARTUP_ORDER=
RSYNC_LINK=0
BACKUP_FILES_CHMOD=
EOF

to save it press esc then :wq then enter

Check the names from the vm's
vim-cmd vmsvc/getallvms

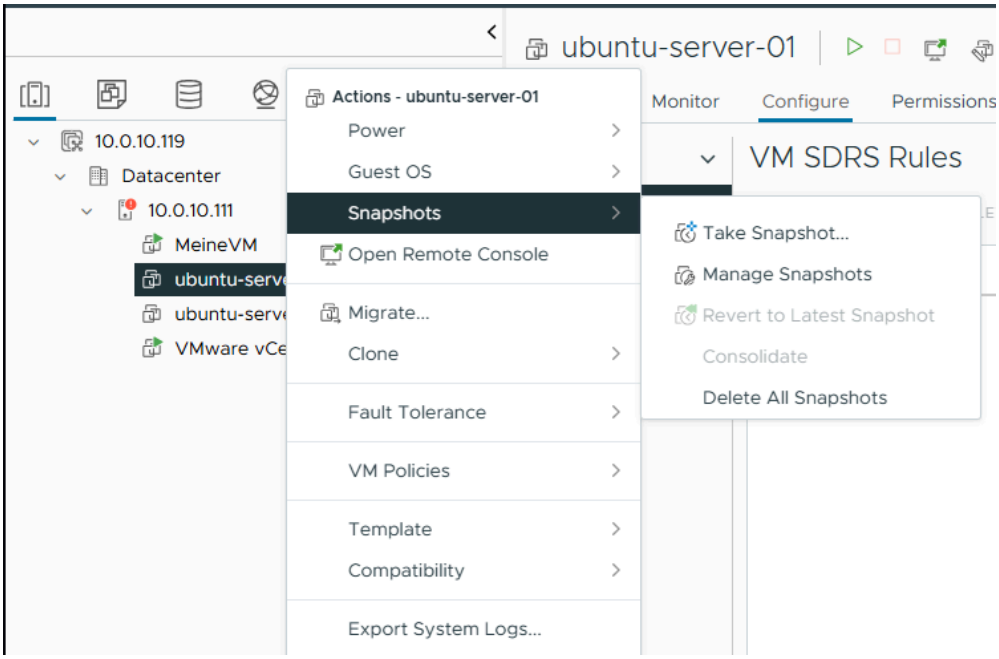
Then execute this command with the vm names
cat > vms_to_backup << 'EOF'
ubuntu-server-01
ubuntu-server-01-clone
EOF
Now we create the Backups with this command
./ghettoVCB.sh -f vms_to_backup -g
/vmfs/volumes/69e5cb3f-c2b3ee88-e01d-d4f5ef8b6f14/ghettoVCB/ghettoVCB.conf

The log output:

2026-05-06 08:42:43 -- info: Successfully completed backup for ubuntu-server-01!
2026-05-06 08:42:50 -- info: Successfully completed backup for ubuntu-server-01-clone!

Create Snapshot (vSphere Client)

Connect to the vSphere Client
Right click on your VM → Snapshot → Take Snapshot



On the Popup Select a Name and maybe a Description → create

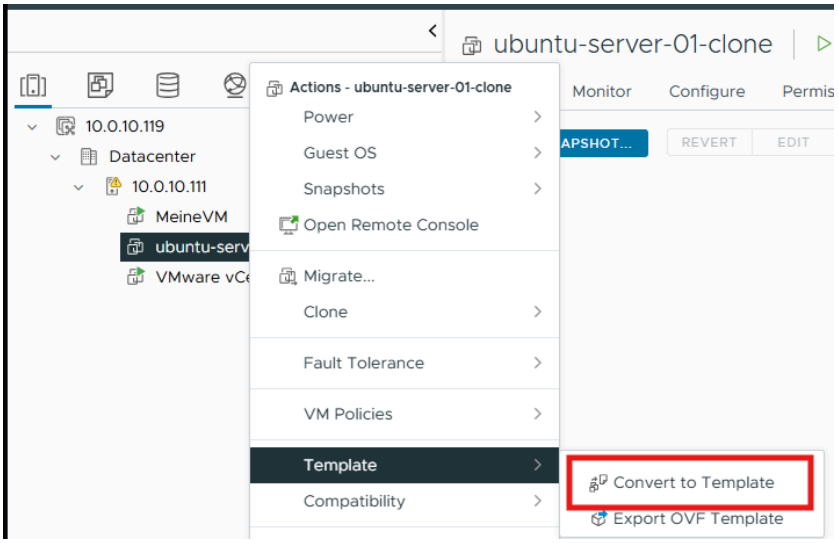
To restore a snapshot Go to Snapshots an than click Revert to Latest Snapshot
On the Popup click Revert

To MAnage teh snapshot Go to Snapshots an then click Manage Snapshot here you can see all your snapshots you can edit and delete them.

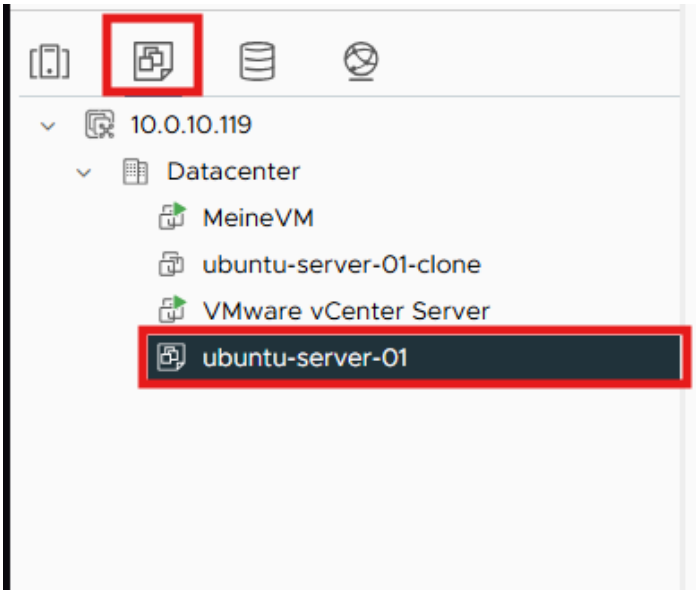
Create Template (vSphere Client)

Warning we make a VM to a Template the vm can’t be used anymore

Rigt click on the vm → Template → convert to Template



Now you find the Template here:



ISCSI

Phase 1: Vorbereitung am TrueNAS

Schritt 1.1: iSCSI-Service prüfen

Im TrueNAS-Web-Interface: **System Settings** → **Services** → bei `iSCSI` muss „Running" und „Start Automatically" aktiv sein.

Schritt 1.2: Bestehendes Setup verifizieren

Unter **Shares** → **Block (iSCSI) Shares Targets** den Tab **Targets** öffnen. Sicherstellen, dass das Target `vmware` existiert und mindestens ein verknüpfter Extent vorhanden ist.

Schritt 1.3: Zvol für ESXi bereitstellen

Unter **Datasets** den Zvol `vmware-esxi` im Pool `areca_raid5` auf eine sinnvolle Größe konfigurieren (in diesem Fall 80 GiB, da der ursprüngliche Wert von 80 MiB zu klein für VMFS war).

> **Hinweis:** Mindestgröße für ein VMFS-Volume liegt bei ~1,3 GB. Bei geteilten NAS-Systemen sollte vor jeder Änderung die Zustimmung des Admins eingeholt und idealerweise ein eigener Zvol mit Benutzer-Präfix angelegt werden.

Schritt 1.4: Daten notieren für die ESXi-Seite

Parameter	Wert
TrueNAS-IP	`10.0.13.3`

iSCSI-Port	`3260`
Target-Basis-IQN	`iqn.2005-10.org.freenas.ctl`
Target-Name	`vmware`
LUN	1 (Zvol `vmware-esxi`, 80 GiB)

Phase 2: Konfiguration am ESXi (vSphere Client)

Schritt 2.1: Software-iSCSI-Adapter aktivieren

1. vSphere Client öffnen (`https://<vcenter>/ui`)
2. **Hosts and Clusters** → ESXi-Host auswählen
3. Tab **Configure** → **Storage Adapters**
4. Button **ADD SOFTWARE ADAPTER** → **Add iSCSI adapter** → **OK**

Ergebnis: Neuer Adapter `vmhbaXX` mit Typ „iSCSI Software Adapter" erscheint in der Liste.

Schritt 2.3: Dynamic Discovery konfigurieren

1. iSCSI-Adapter auswählen
2. Tab **Dynamic Discovery** → **ADD**
3. Felder ausfüllen:
   - iSCSI Server: `10.0.13.3`
   - Port: `3260`
   - Authentication: Inherit from parent (kein CHAP)
4. **OK**

Schritt 2.4: Storage Rescan

Button **RESCAN STORAGE…** klicken → beide Optionen aktiv lassen → **OK**.

Verlauf im Reiter „Recent Tasks" beobachten.

Schritt 2.5: Storage Devices prüfen

Im Configure-Menü links: **Storage Devices**.

Erwartetes Ergebnis: Mindestens eine neue **TrueNAS iSCSI Disk** mit der konfigurierten Größe (80 GiB) und Status „Not Consumed".

Schritt 2.6: VMFS-Datastore anlegen

1. Rechtsklick auf den ESXi-Host → **Storage** → **New Datastore…**
2. Type: **VMFS** → Next

3. Name: z. B. `iSCSI-Datastore-01`, Host auswählen, dann die TrueNAS-LUN markieren → Next
4. VMFS Version: **VMFS 6** → Next
5. Partition Configuration: Use all available partitions → Next
6. **Finish**

Schritt 2.7: Funktionstest mit Storage-Migration

1. Rechtsklick auf eine bestehende VM → **Migrate…**
2. Migration Type: **Change storage only** → Next
3. Ziel-Datastore: `iSCSI-Datastore-01` auswählen → Next → **Finish**
4. Ergebnis im Reiter „Recent Tasks" und in der VM-Übersicht (Datastore-Spalte) verifizieren

---

Verifikation

Prüfpunkt	Erwartetes Ergebnis
TrueNAS → iSCSI Connections	Aktive Verbindung des ESXi-Initiators sichtbar
ESXi → Storage Adapters	iSCSI Software Adapter aktiv
ESXi → Storage Devices	TrueNAS iSCSI Disk mit 80 GiB sichtbar
ESXi → Datastores	`iSCSI-Datastore-01` mit Kapazität sichtbar
VM-Migration	VM-Dateien liegen auf dem neuen Datastore

Vm Storage Migration

You can Stop the VM but you didn’t need it
right click the vm → Migrate → Change Storage only → select new Storage → Finish

Vm Migration Hypervisor

ADD second Hypervisor

on the Datacenter rightclick → ADD Host

Wizard durchklicken:

- **Name and location:** IP oder FQDN des zweiten ESXi → Next
- **Connection settings:** User `root` + Passwort → Next
- **Security Alert:** Zertifikat akzeptieren → Yes
- **Host summary:** prüfen → Next
- **Assign license:** Eval lassen → Next
- **Lockdown mode:** Disabled → Next
- **VM location:** Datacenter → Next
- **Ready to complete** → **Finish**

Shared Storage

Step 1: Select the Second Host

In the vSphere Client inventory pane, click on the second ESXi host. Make sure not to select the first host, as configuration changes would otherwise be applied to the wrong target.

Step 2: Enable the Software iSCSI Adapter

1. Open the **Configure** tab at the top.
2. In the left-hand menu, navigate to **Storage** → **Storage Adapters**.
3. Click the **ADD SOFTWARE ADAPTER** button and select **Add iSCSI adapter**, then confirm with **OK**.

A new adapter (typically named `vmhbaXX`, type "iSCSI Software Adapter") will appear in the list.

Step 3: Configure Dynamic Discovery

1. Select the newly created iSCSI adapter.
2. In the details pane at the bottom, open the **Dynamic Discovery** tab and click **ADD**.
3. Enter the following values:
    - **iSCSI Server:** `10.0.13.3`
    - **Port:** `3260`
4. Confirm with **OK**.

Step 4: Trigger a Storage Rescan

In the Storage Adapters view, click the **RESCAN STORAGE…** button at the top. Leave both checkboxes enabled and confirm with **OK**.

Step 5: Verify Storage Devices

In the left-hand menu, navigate to **Storage Devices**.

The same TrueNAS iSCSI disks that were visible on the first host should now appear here as well (device: 80 GB.)

vMotion Network

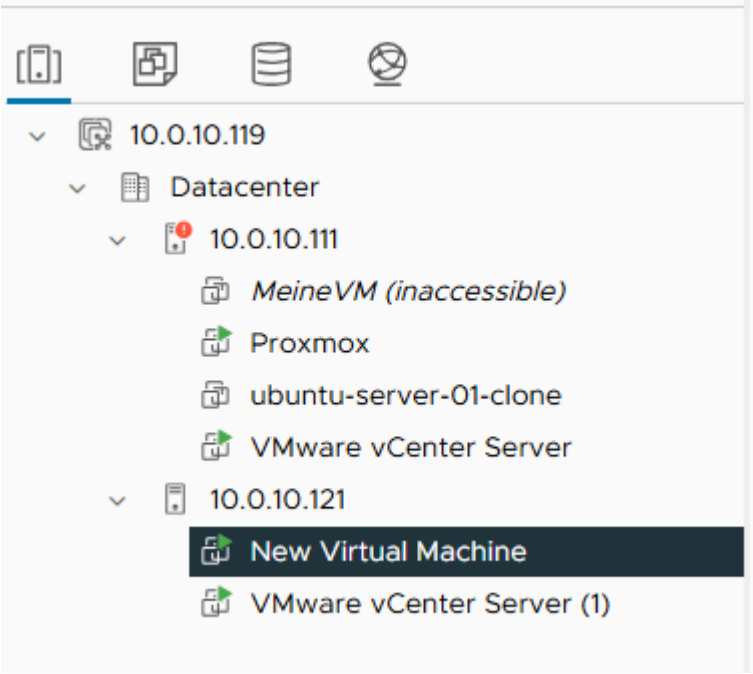
This must be done on every Host

Host → **Configure** → **VMkernel adapters**
click on the three dots → edit → enable vMotion

Migrate Vm

Requirements:
Vm need a Shared Storage
ISO must be disconnected

Right Click the vm → migrate
Wizard:
Select a migration type: Change compute resource only
Select a compute resource: Select new Host
Select networks: Next
Select vMotion priority: Next
Ready to complete: Finish

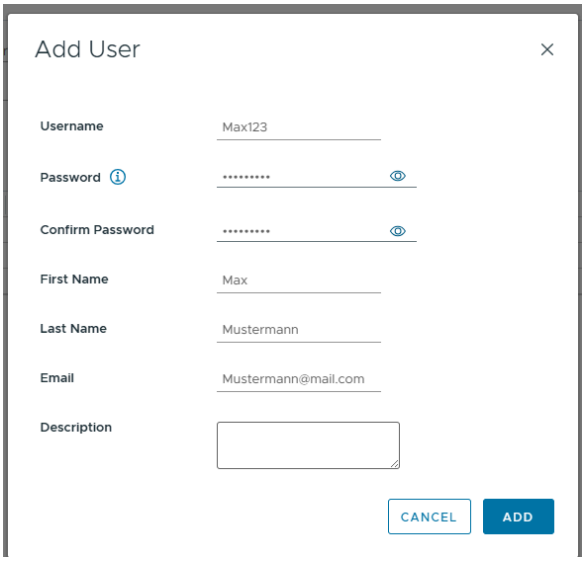


Users & Groups

Open ☰ → Administration
Close ☰ under Single Sign On → Users and Groups

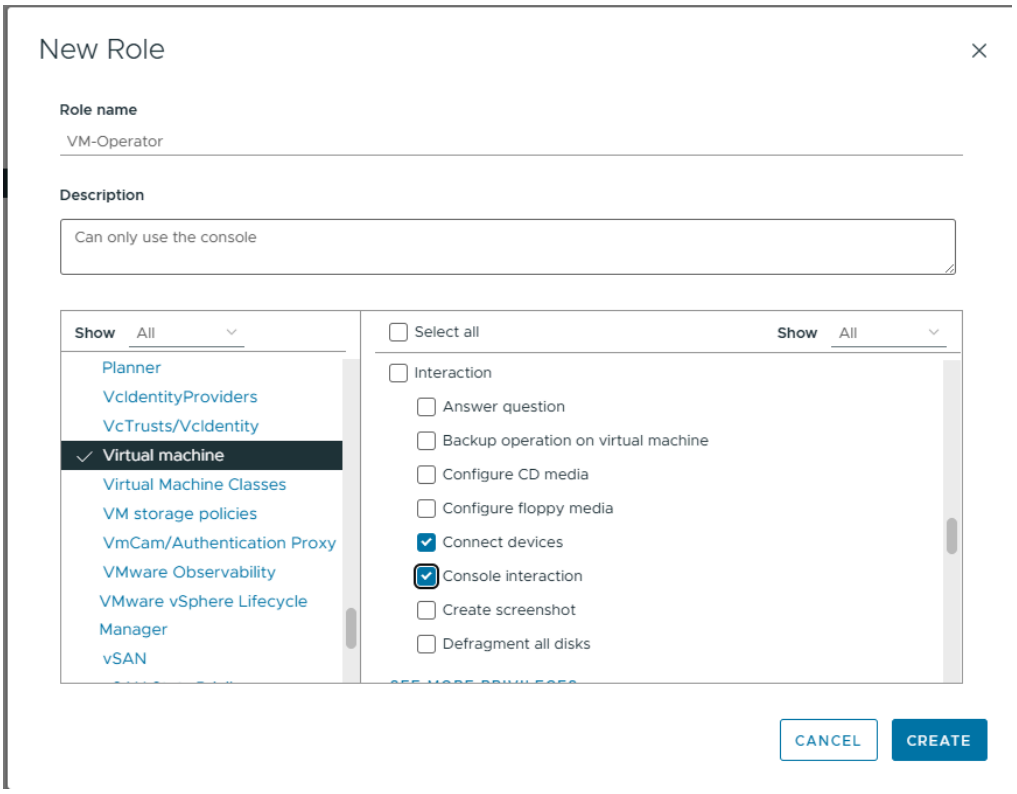
Domain : vsphere.local

ADD → Create User



Create Roles

In the Left-Sidebar → Roles → New



Create

Set Permissions

On Host & Clusters select a Datacenter or a VM
right click an select add Permission
in the Wizzard:
Domain: vsphere.local
User/Groups: select new Video
Role: VM-Operator
Propagate to children: If you turn it on e.x. on a datacenter, all vm's become this rules

Step 1: Open the vSphere Lifecycle Manager

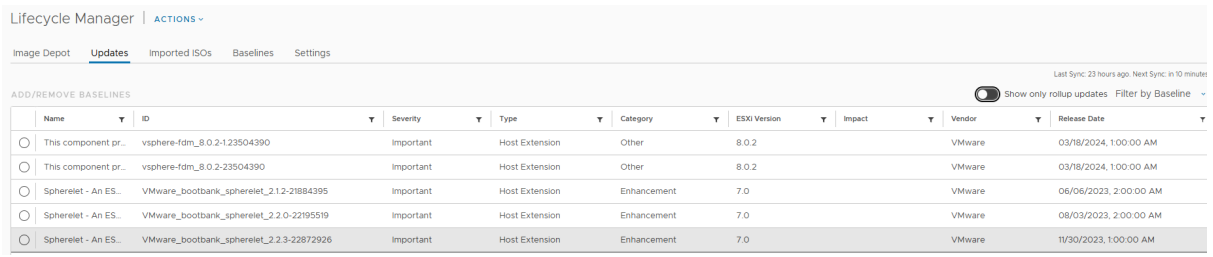
In the vSphere Client, navigate to **Menu → Lifecycle Manager**.

Step 2: Verify Patch Source

Open the **Settings → Patch Downloads** tab and confirm that the VMware Online Depot is configured and reachable. If required, trigger a manual sync using **Sync Updates**.

Step 3: Attach a Baseline to the Host

1. Navigate to **Hosts and Clusters → [ESXi Host] → Updates → Baselines**.
2. Click **ATTACH** and select the predefined baseline **Critical Host Patches (Predefined)**.



Step 4: Run Compliance Check

Click **CHECK COMPLIANCE**. The host is compared against the attached baseline.

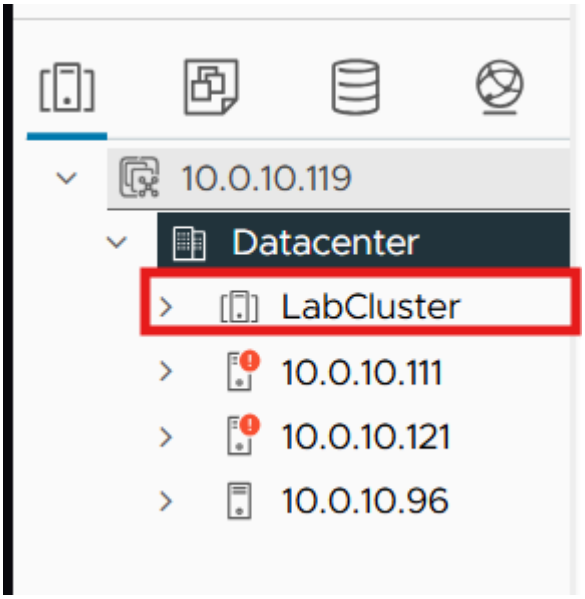
Result: The host was reported as **Compliant**, indicating that no updates from the baseline were missing.

Clustering

vSphere Client → Hosts and Clusters
Right click on your Datacenter → New Cluster…
Wizard:
Name → LabCluster
vSphere HA → True

Manage all hosts in the cluster with a single image → False

Next → Finish



Connect Host WIth the Cluster

Via Drag & Drop you can drop your Hosts over the Cluster to add them.

Fine-Tuning HA Settings

Select the cluster → **Configure** → **Services** → **vSphere Availability** → **EDIT…**

Failures and Responses

- **Host failure response:** Restart VMs
- **Response for host isolation:** Power off and restart VMs
- **Datastore with PDL:** Power off and restart VMs
- **Datastore with APD:** Power off and restart VMs (conservative)
- **VM Monitoring:** VM Monitoring Only

Admission Control

- **Host failures cluster tolerates:** 1

Heartbeat Datastores

- **Selection policy:** Automatically select datastores accessible from the host

Apply

Click **OK**. vCenter pushes the HA configuration to all hosts (visible in Recent Tasks). Process completes within 1–2 minutes.

Adding a Second Heartbeat Datastore

Background

After enabling vSphere HA on the cluster, the following warning appeared on each host:

> *"The number of vSphere HA heartbeat datastores for this host is 1, which is less than required: 2"*

vSphere HA uses datastores as a **secondary communication channel** between hosts, in addition to the management network. These so-called **heartbeat datastores** allow the HA master to distinguish between two very different failure scenarios:

- **A host has actually failed** (no network heartbeats, no datastore heartbeats) — HA restarts the VMs on other hosts.
- **A host is only network-isolated** but still running (no network heartbeats, but datastore heartbeats still present) — HA reacts more cautiously to avoid a split-brain situation where the same VM would run on two hosts.

To make this fallback reliable, vSphere HA requires **at least two heartbeat datastores** that are accessible by all hosts in the cluster. If only one is available and that datastore becomes unreachable, HA loses its ability to correctly assess host states.

Initially, the cluster had only one shared datastore (`iSCSI-Datastore-01`, 78 GB), so a second one had to be added.

Procedure

Step 1: Create a New Zvol on TrueNAS

- **Datasets → areca_raid5 → Add Zvol**
- **Name:** `max-heartbeat`
- **Size:** `5 GiB` (sufficient for heartbeat purposes only)
- **Sparse:** enabled (thin-provisioned)

Step 2: Create a New Extent

- **Shares → Block (iSCSI) Shares Targets → Extents → Add**
- **Name:** `max-heartbeat-extent`
- **Extent Type:** Device
- **Device:** `zvol/areca_raid5/max-heartbeat`
- **Sharing Platform:** *VMware: Extent block size 512b, TPC enabled, no Xen compat mode, SSD speed*

Step 3: Associate the Extent with the iSCSI Target

In the Target `vmware` configuration, add a new LUN association:

- **LUN ID:** `2`
- **Extent:** `max-heartbeat-extent`

Step 4: Rescan iSCSI Adapter on All Hosts

On each of the three ESXi hosts:

Configure → Storage Adapters → iSCSI Adapter → RESCAN STORAGE

The new 5 GiB LUN appears under **Storage Devices** with status "Not Consumed".

Step 5: Create the VMFS Datastore

- Right-click any host → **Storage** → **New Datastore**
- **Type:** VMFS
- **Name:** `iSCSI-Heartbeat`
- **Disk:** the new 5 GiB TrueNAS LUN
- **Version:** VMFS 6

The new datastore is automatically mounted on all three hosts because the underlying target is shared.

Step 6: Reconfigure HA to Use Both Datastores

- Cluster → **Configure** → **vSphere Availability** → **EDIT** → **Heartbeat Datastores**
- **Policy:** *Use datastores from the specified list and complement automatically if needed*
- Select both shared datastores: `iSCSI-Datastore-01` and `iSCSI-Heartbeat`
- Apply

Result

After HA was reconfigured, **Monitor → vSphere HA → Heartbeat** showed two heartbeat datastores per host, and the warning disappeared. The cluster is now resilient against the failure of a single heartbeat datastore: if one becomes unreachable, the second one continues to provide secondary failure-detection signaling.

Why This Matters

A single heartbeat datastore is a **single point of failure** for HA's failure-detection logic. Without redundancy, HA can no longer reliably distinguish between an actual host failure and a transient network or storage issue, which can lead to incorrect failover decisions, unnecessary VM restarts, or — worse — failed restarts when they are actually needed. Maintaining at least two heartbeat datastores is therefore a fundamental requirement for any production HA cluster.