

Virtuelle Workstation mit Proxmox und mehreren Screens via X11

1. Grundlegende Architektur

Komponente	Beschreibung
Proxmox-Host (Hetzner)	Ryzen 12-Core-Server mit Proxmox VE, UFW-Firewall (nur SSH & Proxmox-Ports).
Windows-VM	VM mit 2 virtuellen Monitoren , erreichbar über RDP/NoMachine .
Linux-VM	VM mit 2 virtuellen Monitoren , erreichbar über NoMachine/x2go .
Firewall (UFW)	Muss RDP (3389) , NoMachine (4000) , x2go (22/SSH) freigeben.
Zugang über Internet	Reverse-Proxy (optional) für sicheren Zugriff oder WireGuard-VPN .

2. Schritt-für-Schritt-Anleitung

☐☐ Schritt 1: VMs erstellen (Windows & Linux)

Windows-VM (mit 2 virtuellen Monitoren)

1. VM in Proxmox erstellen:

- **OS:** Windows 10/11 (ISO von [Microsoft](#)).
 - **CPU:** 4-6 Kerne (Ryzen 12-Core erlaubt das).
 - **RAM:** 8-16 GB (je nach Nutzung).
 - **Grafik: SPICE (QXL)** oder **Standard-VGA** (wir nutzen später **RDP/NoMachine**).
 - **Festplatte:** 50-100 GB (SSD/NVMe empfohlen).
 - **Netzwerk: Bridge (vmbro)** für Internetzugriff.
2. **Windows installieren & Treiber aktualisieren:**

- **VirtIO-Treiber** installieren ([Download](#)).
- **SPICE-Guest-Tools** installieren (für bessere Grafikperformance):

```
# In der VM (als Admin):  
msiexec /i spice-guest-tools-latest.exe
```

- **2 virtuelle Monitore aktivieren** (über **RDP** oder **NoMachine**).
-

Linux-VM (mit 2 virtuellen Monitoren, z. B. Ubuntu/KDE)

1. **VM in Proxmox erstellen:**

- **OS:** Ubuntu 22.04 LTS oder Debian 12 (mit **KDE/XFCE** für bessere Multi-Monitor-Unterstützung).
- **CPU:** 4-6 Kerne.
- **RAM:** 4-8 GB.
- **Grafik: SPICE (QXL)** oder **VirtIO-GPU** (für bessere Performance).
- **Festplatte:** 30-50 GB.
- **Netzwerk: Bridge (vmbro)**.

2. **Linux installieren & für Multi-Monitor vorbereiten:**

- **SPICE-Treiber installieren:**

```
sudo apt update && sudo apt install spice-vdagent spice-webdavd
```

- **X11/KDE für 2 Monitore konfigurieren:**

```
# Virtuellen Monitor hinzufügen (z. B. rechts vom Hauptbildschirm)  
xrandr --output Virtual1 --mode 1920x1080 --right-of eDP1
```

(Für **permanente Einstellung** in `~/.xprofile` eintragen.)

☐☐ Schritt 2: Remote-Zugriff einrichten (RDP/NoMachine/x2go)

Option 1: RDP (für Windows-VM)

- **Vorteil:** Einfach, integriert, Multi-Monitor-Unterstützung.

- **Einrichtung:**

1. **RDP in Windows aktivieren:**

- Win + R → `sysdm.cpl` → "**Remoteverbindungen zulassen**".

2. **Port in UFW freigeben:**

```
sudo ufw allow 3389/tcp comment "RDP for Windows VM"
```

3. **Von außen verbinden:**

- **Mit** `mstsc.exe /multimon` (Windows) oder **Remmina** (Linux).
- **IP:Port** = `DEIN_SERVER_IP:3389` (oder **Port-Forwarding**, falls NAT im Spiel ist).

⚠ Sicherheitshinweis:

- **RDP ist unsicher, wenn direkt dem Internet ausgesetzt!**
- **Besser: WireGuard-VPN** oder **Reverse-Proxy mit Authentifizierung** (z. B. **Nginx + Authelia**).

Option 2: NoMachine (für Windows & Linux, besser als RDP)

- **Vorteil:** Schnell, gute Multi-Monitor-Unterstützung, verschlüsselt.

- **Einrichtung:**

1. **NoMachine auf VMs installieren:**

- Download (für Windows & Linux).
- **Server installieren** (Standard-Port: **4000**).

2. **Port in UFW freigeben:**

```
sudo ufw allow 4000/tcp comment "NoMachine"
```

3. **Von außen verbinden:**

- **NoMachine-Client** installieren → `DEIN_SERVER_IP:4000` eingeben.
- **2 Monitore aktivieren** in den NoMachine-Einstellungen.

☑ **Sicherer als RDP!** (Verschlüsselung, weniger Angriffsfläche).

Option 3: x2go (für Linux, leichtgewichtig)

- **Vorteil:** Gut für Linux, nutzt **SSH (Port 22)**.

- **Einrichtung:**

1. **x2go-Server installieren:**

```
sudo apt install x2goserver x2goserver-xsession
```

2. **Port 22 ist bereits offen (SSH).**
3. **Von außen verbinden:**
 - **x2go-Client** installieren → `DEIN_SERVER_IP` + **SSH-Benutzer** eingeben.
 - **2 Monitore** in den Sitzungseinstellungen aktivieren.

☐☐ Schritt 3: Firewall (UFW) & Sicherheit

Aktuell lässt deine UFW nur **SSH & Proxmox-Ports** zu. Du musst **zusätzliche Ports freigeben**:

Dienst	Port	Befehl (UFW)	Sicherheitsrisiko
RDP	3389	<code>sudo ufw allow 3389/tcp</code>	⚠ Hoch
NoMachine	4000	<code>sudo ufw allow 4000/tcp</code>	☐ Mittel
x2go	22	(bereits offen)	☐ Niedrig (SSH)

☐☐ Sicherheitsempfehlungen

1. **WireGuard-VPN einrichten** (beste Lösung für sicheren Remote-Zugriff):
 - **Tutorial:** [WireGuard auf Proxmox](#).
 - **Vorteil:** Nur VPN-Benutzer können auf RDP/NoMachine zugreifen.
2. **Fail2Ban installieren** (gegen Brute-Force):

```
sudo apt install fail2ban
sudo systemctl enable fail2ban
```

3. **Port-Knocking** (optional, für zusätzliche Sicherheit).
4. **Reverse-Proxy mit Authentifizierung** (z. B. **Nginx + Authelia**):
 - Beispiel: `https://deine-domain.com/rdp` → leitet zu RDP weiter, aber mit **2FA**.

☐☐ Schritt 4: Performance-Optimierung

- **Windows-VM:**
 - **GPU-Passthrough** (falls möglich, aber bei Hetzner **nicht erlaubt**).

- **Alternativ: VirtIO-GPU** (bessere Performance als Standard-VGA).

```
# In der Proxmox-VM-Konfiguration (/etc/pve/qemu-server/<VMID>.conf):  
args: -device virtio-gpu-pci
```

- **Linux-VM:**

- **SPICE + VirtIO-GPU** für bessere Grafik:

```
# In der VM-Konfig:  
vga: virtio
```

3. Fertige Lösung: So sieht der Workflow aus

1. **Du startest die Windows-VM** (z. B. für Office/Adobe).
 - Verbindung über **NoMachine/RDP** mit **2 Monitoren**.
2. **Du startest die Linux-VM** (z. B. für Entwicklung/Docker).
 - Verbindung über **NoMachine/x2go** mit **2 Monitoren**.
3. **UFW schützt die Ports**, aber:
 - **WireGuard-VPN** (empfohlen) oder **Reverse-Proxy** macht es sicherer.
4. **Beide VMs laufen gleichzeitig**, aber nur eine wird genutzt (Ressourcen-sparend).

4. Alternative: Einfacherer Ansatz mit "Single-Port-Lösung"

Falls du **keine zusätzlichen Ports öffnen willst**, kannst du **alles über SSH tunneln**:

```
# Beispiel: RDP über SSH-Tunnel (lokaler Port 3390 → VM:3389)  
ssh -L 3390:localhost:3389 user@DEIN_SERVER_IP
```

Dann verbindest du dich mit `localhost:3390` (nur lokal, sicher).

Vorteil: Keine zusätzlichen Firewall-Regeln nötig!

5. Zusammenfassung: Was du tun musst

Schritt	Aktion
☐ VMs erstellen	Windows & Linux-VM mit SPICE/VirtIO-GPU.
☐ Multi-Monitor einrichten	Windows: RDP/NoMachine, Linux: xrandr/NoMachine.
☐ Remote-Zugriff konfigurieren	RDP (3389), NoMachine (4000), x2go (22).
☐ Firewall anpassen	<code>sudo ufw allow 3389,4000/tcp</code> .
☐ Sicherheit erhöhen	WireGuard-VPN oder Reverse-Proxy.
☐ Performance optimieren	VirtIO-GPU & SPICE-Treiber.

6. Empfohlene Tools für besseres Erlebnis

Tool	Zweck	Link
NoMachine	Schnellster Remote-Desktop mit Multi-Monitor	nomachine.com
x2go	Leichtgewichtig für Linux	x2go.org
Remmina	RDP-Client für Linux	remmina.org
WireGuard	Sicheres VPN für Remote-Zugriff	wireguard.com
Authelia	2FA für Reverse-Proxy	authelia.com

Fazit: Ja, das geht!

☐ Ja, du kannst auf deinem Hetzner-Proxmox-Server:

- **Eine Windows-VM mit 2 virtuellen Monitoren** (über RDP/NoMachine).
- **Eine Linux-VM mit 2 virtuellen Monitoren** (über NoMachine/x2go).
- **Beide über das Internet erreichbar** (mit UFW & Sicherheitseinstellungen).
- **Nur eine VM gleichzeitig nutzen** (Ressourcen-sparend).

☐ **Beste Lösung für Sicherheit: WireGuard-VPN + NoMachine** (statt RDP).

☐ **Beste Lösung für Einfachheit: NoMachine (Port 4000) + UFW-Regel.**

Fragen?

- Brauchst du Hilfe bei der **UFW-Konfiguration**?
 - Soll ich eine **WireGuard-Anleitung** für Proxmox detaillierter beschreiben?
 - Oder möchtest du eine **Step-by-Step für NoMachine**?
-

Revision #1

Created 21 September 2025 06:52:28 by Stefan Mechler

Updated 21 September 2025 06:54:05 by Stefan Mechler