

# Virtuelle Workstation mit Proxmox und mehreren Screens via Wayland

Ja, **Wayland** macht die Sache etwas komplizierter, aber es gibt **Lösungen**, um **virtuelle Multi-Monitor-Setups in einer Proxmox-VM mit Wayland** (z. B. Kubuntu 24.04+) zu realisieren. Da Wayland **kein** `xrandr` unterstützt und **klassische Remote-Tools** (wie RDP oder x2go) oft noch auf X11 angewiesen sind, müssen wir **Workarounds** nutzen.

## 1. Wayland & virtuelle Monitore: Die Herausforderungen

Problem	Erklärung
Kein <code>xrandr</code>	Wayland verwaltet Monitore über <b>Kompositoren</b> (KWin, Mutter, Weston), nicht über X11-Tools.
Kein direkter Multi-Monitor-Support in SPICE/QXL	Die standardmäßige <b>SPICE-Grafikkarte</b> in Proxmox/QEMU unterstützt Wayland nur eingeschränkt.
RDP/NoMachine/x2go oft X11-basiert	Viele Remote-Tools setzen auf X11 und funktionieren nicht nativ mit Wayland.
Wayland-Sessions blockieren Remote-Zugriff	Standardmäßig erlaubt Wayland <b>keine Remote-Sessions</b> (Sicherheitsfeature).

# 2. Lösungen für Wayland in Proxmox-VMs

## ☐☐ Option 1: Wayland mit "virtuellen Monitoren" über KWin (KDE)

### Funktionsweise

KDEs **KWin** (Wayland-Kompositor) erlaubt das **Hinzufügen virtueller Monitore** über **Skripte oder manuelle Konfiguration**.

### Schritt-für-Schritt-Anleitung (Kubuntu 24.04+)

#### 1. VM erstellen (Kubuntu 24.04 mit Wayland)

- **Grafik: VirtIO-GPU** (besser als SPICE für Wayland).

```
# In /etc/pve/qemu-server/<VMID>.conf:  
vga: virtio
```

- **CPU/RAM:** Mind. 4 Kerne, 8 GB RAM (Wayland braucht mehr Ressourcen als X11).

#### 2. KWin für virtuelle Monitore konfigurieren

- **Manuell über `kwin_wayland`:**

```
# Temporärer virtueller Monitor (rechts vom Hauptbildschirm)  
kwin_wayland --xwayland --width 3840 --height 1080
```

- **Permanent über Autostart:**

- Erstelle eine Datei `~/config/autostart/kwin-virtual-monitor.desktop`:

```
[Desktop Entry]  
Name=KWin Virtual Monitor  
Exec=kwin_wayland --xwayland --width 3840 --height 1080  
Type=Application
```

- **Alternativ: `wlr-randr` (für wlroots-basierte Compositors)**

```
sudo apt install wlr-randr  
wlr-randr --output Virtual1 --mode 1920x1080 --right-of DP-1
```

#### 3. Remote-Zugriff einrichten (NoMachine oder RDP über XWayland)

- **NoMachine** funktioniert mit **XWayland** (X11-Kompatibilitätsschicht in Wayland).
  - Installiere NoMachine wie gehabt, aber starte die Session im **XWayland-Modus**.
- **RDP (xrdp) mit XWayland:**

```
sudo apt install xrdp
sudo systemctl enable --now xrdp
```

- **Problem:** Standardmäßig startet `xrdp` eine **X11-Session**, nicht Wayland.
- **Lösung: Manuell Wayland-Session erzwingen** (experimentell):

```
# /etc/xrdp/startwm.sh anpassen:
export XDG_SESSION_TYPE=wayland
export QT_QPA_PLATFORM=wayland
exec dbus-run-session -- startplasma-wayland
```

---

## ☐☐ Option 2: Wayland mit SPICE + spice-vdagent (begrenzt)

### Funktionsweise

- **SPICE** (Proxmox-Standard) unterstützt **Wayland nur eingeschränkt**, aber mit `spice-  
vdagent` kann man **Bildschirmgrößen anpassen**.
- **Keine echten virtuellen Monitore**, aber **ein großer Desktop, der wie zwei Monitore aussieht**.

### Einrichtung

#### 1. SPICE-Treiber installieren:

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

#### 2. VM-Konfiguration anpassen ( `/etc/pve/qemu-server/<VMID>.conf` ):

```
args: -device virtio-gpu-pci -display spice,gl=on
```

#### 3. Manuell "Fake-Multi-Monitor" einrichten:

- Setze die **Auflösung auf 3840x1080** (2x 1920x1080 nebeneinander).
- Nutze **KWin-Skripting**, um den Desktop in zwei Hälften zu teilen:

```
kwriteconfig5 --file kwinrc --group Windows --key BorderlessMaximizedWindows true
```

- **Nachteil:** Keine echte Multi-Monitor-Unterstützung (nur ein großer Bildschirm).

---

# ☐☐ Option 3: Wayland mit NoMachine (beste Lösung für Remote)

## Funktionsweise

- **NoMachine** unterstützt **Wayland seit Version 8** (aber noch experimentell).
- **Vorteil:** Echte **Multi-Monitor-Unterstützung** über das Netzwerk.

## Einrichtung

### 1. NoMachine installieren:

```
wget https://download.nomachine.com/download/8.6/Linux/nomachine_8.6.1_1_amd64.deb
sudo dpkg -i nomachine_*.deb
```

### 2. Wayland-Session erzwingen:

- Bearbeite `/usr/NX/etc/node.cfg`:

```
EnableWaylandSupport 1
```

### 3. Neu starten & verbinden:

- NoMachine-Client → **Wayland-Session auswählen**.
- **2 Monitore** in den Einstellungen aktivieren.

---

# ☐☐ Option 4: XWayland als Fallback (einfachste Lösung)

## Funktionsweise

- **XWayland** ist eine **X11-Kompatibilitätsschicht** in Wayland.
- **Vorteil:** Du kannst **X11-Tools wie `xrandr`** nutzen, während die Session eigentlich Wayland ist.

## Einrichtung

1. **XWayland aktivieren** (standardmäßig aktiv in Kubuntu).
2. **Virtuelle Monitore mit `xrandr` hinzufügen** (auch in Wayland!):

```
xrandr --output Virtual1 --mode 1920x1080 --right-of eDP1
```

3. Remote-Tools nutzen (RDP, NoMachine, x2go) über XWayland.

## 3. Remote-Zugriff für Wayland (Sicher & Performant)

Methode	Wayland-Unterstützung	Multi-Monitor	Sicherheit	Empfehlung
<b>NoMachine (Wayland-Modus)</b>	☑ (seit v8)	☑	████	<b>Beste Wahl</b>
<b>RDP (xrdp + XWayland)</b>	△ (X11-Fallback)	☑	███	Gut für Windows-Clients
<b>SPICE + VirtIO-GPU</b>	△ (kein echter Multi-Monitor)	☑	████	Nur für Single-Monitor
<b>VNC (TigerVNC)</b>	☑ (nur X11)	△	██	Nicht ideal
<b>WireGuard + NoMachine</b>	☑	☑	█████	<b>Sicherste Lösung</b>

## 4. Schritt-für-Schritt: Kubuntu 24.04 in Proxmox mit Wayland & 2 Monitoren

### ██ Vorbereitung (Proxmox-Host)

1. VM erstellen:

- **OS:** Kubuntu 24.04 ISO.
- **Grafik:** **VirtIO-GPU** (nicht SPICE!).

```
# In /etc/pve/qemu-server/<VMID>.conf:  
vga: virtio
```

- **CPU:** 4 Kerne, **RAM:** 8 GB.

## 2. **VirtIO-Treiber installieren** (für bessere Performance):

```
sudo apt install virtio-driver
```

---

# ☐☐ Einrichtung in der VM (Kubuntu 24.04)

1. **Wayland-Session starten** (standardmäßig in Kubuntu 24.04).
2. **Virtuelle Monitore hinzufügen** (über KWin):

```
# Temporärer Test:  
kwin_wayland --xwayland --width 3840 --height 1080
```

- **Permanent:**

- Erstelle ein **KWin-Skript** (`~/.config/kwinrc`):

```
[ScreenEdges]  
BorderlessMaximizedWindows=true
```

- **Autostart-Skript** für virtuelle Monitore:

```
echo 'kwin_wayland --xwayland --width 3840 --height 1080' >> ~/.config/autostart/kwin-  
virtual.sh  
chmod +x ~/.config/autostart/kwin-virtual.sh
```

## 3. **NoMachine für Remote-Zugriff installieren:**

```
wget https://download.nomachine.com/download/8.6/Linux/nomachine_8.6.1_1_amd64.deb  
sudo dpkg -i nomachine_*.deb  
sudo systemctl enable --now nxserver
```

- **Konfiguration für Wayland:**

```
sudo sed -i 's/EnableWaylandSupport 0/EnableWaylandSupport 1/' /usr/NX/etc/node.cfg  
sudo systemctl restart nxserver
```

## 4. **Firewall (UFW) anpassen:**

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

## ☐☐ Verbindung von außen (Client)

1. **NoMachine-Client installieren** ([Download](#)).
2. **Verbindung herstellen:**
  - **Server:** `DEIN_SERVER_IP:4000`
  - **Session-Typ:** "Wayland" auswählen.
  - **2 Monitore aktivieren** in den Einstellungen.

# 5. Sicherheitsoptimierung (wichtig für Internet-Zugriff!)

## ☐☐ Option A: WireGuard-VPN (empfohlen)

- **Vorteil:** Keine offenen Ports im Internet, nur VPN-Zugriff.
- **Einrichtung:**

```
# Auf dem Proxmox-Host:  
sudo apt install wireguard  
wg genkey | tee privatekey | wg pubkey > publickey
```

- **Konfiguration:** `/etc/wireguard/wg0.conf`

```
[Interface]  
PrivateKey = <SERVER_PRIVATE_KEY>  
Address = 10.0.0.1/24  
ListenPort = 51820  
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o vmbr0 -j  
MASQUERADE  
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o vmbr0 -j  
MASQUERADE
```

```
[Peer]
PublicKey = <CLIENT_PUBLIC_KEY>
AllowedIPs = 10.0.0.2/32
```

◦ **UFW-Regel:**

```
sudo ufw allow 51820/udp comment "WireGuard"
```

◦ **Client-Konfiguration** (z. B. auf deinem Laptop):

```
[Interface]
PrivateKey = <CLIENT_PRIVATE_KEY>
Address = 10.0.0.2/24

[Peer]
PublicKey = <SERVER_PUBLIC_KEY>
Endpoint = DEIN_SERVER_IP:51820
AllowedIPs = 10.0.0.0/24
```

◦ **Verbindung herstellen:**

```
sudo wg-quick up wg0
```

◦ **Dann NoMachine/RDP über VPN nutzen** ( `10.0.0.1:4000` ).

---

## ☐☐ Option B: Reverse-Proxy mit Authelia (für direkte Exposition)

- **Vorteil:** Kein VPN nötig, aber **2FA & HTTPS**.
- **Einrichtung (Nginx + Authelia):**

```
sudo apt install nginx certbot python3-certbot-nginx
sudo snap install --edge authelia
```

◦ **Nginx-Konfiguration** ( `/etc/nginx/sites-available/nomachine` ):

```
server {
    listen 443 ssl;
    server_name nomachine.deine-domain.com;
```

```

ssl_certificate /etc/letsencrypt/live/nomachine.deine-domain.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/nomachine.deine-domain.com/privkey.pem;

location / {
    proxy_pass http://localhost:4000;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_set_header Host $host;
}
}

```

o **Authelia für 2FA:**

```

# /etc/authelia/configuration.yml
authentication_backend:
  file:
    path: /etc/authelia/users_database.yml
access_control:
  default_policy: deny
  rules:
    - domain: "nomachine.deine-domain.com"
      policy: two_factor

```

o **Zertifikat mit Let's Encrypt:**

```

sudo certbot --nginx -d nomachine.deine-domain.com

```

## 6. Performance-Tipps für Wayland in Proxmox

Problem	Lösung
<b>Laggy Grafik</b>	Nutze <b>VirtIO-GPU</b> statt SPICE.
<b>Keine Hardware-Beschleunigung</b>	Aktiviere <b>OpenGL in NoMachine</b> : <code>sudo nxserver --enable-opengl</code>
<b>Wayland-Session crasht</b>	Starte mit <b>XWayland-Fallback</b> : <code>export XDG_SESSION_TYPE=x11</code>

Problem	Lösung
Keine virtuellen Monitore	Nutze <b>KWin-Skripting</b> oder <code>wlr-randr</code> .
Hohe CPU-Last	Reduziere die <b>Auflösung</b> (z. B. 1920x1080 statt 4K).

## 7. Fazit: Wayland in Proxmox mit Multi-Monitor

Methode	Funktionsweise	Vorteile	Nachteile
<b>NoMachine (Wayland-Modus)</b>	Echte Wayland-Session mit Multi-Monitor	☐ Beste Performance, ☐ Echte Wayland-Unterstützung	△ Noch experimentell
<b>XWayland + xrandr</b>	Wayland mit X11-Kompatibilität	☐ Einfach, ☐ Stabil	☐ Kein "echtes" Wayland
<b>KWin-Skripting</b>	Virtuelle Monitore über KWin	☐ Nativ in Wayland	△ Manuelle Konfiguration nötig
<b>SPICE + VirtIO-GPU</b>	Emulierte Grafik	☐ Einfach	☐ Kein echter Multi-Monitor

### ☐☐ Empfohlene Lösung für dein Setup

1. **Kubuntu 24.04-VM mit VirtIO-GPU.**
2. **NoMachine in Wayland-Modus** (Port 4000).
3. **WireGuard-VPN** für sicheren Zugriff.
4. **KWin-Skript für virtuelle Monitore** (falls NoMachine nicht reicht).

### ☐☐ Alternative: Bei X11 bleiben (bis Wayland stabiler ist)

Falls Wayland zu viele Probleme macht:

```
# X11 erzwingen (in Kubuntu 24.04)
echo "exec startplasma-x11" > ~/.xsession
```

- **Vorteil:** `xrandr` funktioniert, NoMachine/RDP/x2go laufen stabil.

- **Nachteil:** Kein Wayland (aber für Remote-Setups oft besser).
- 

# 8. Zusammenfassung: Was du tun musst

Schritt	Aktion
☐ <b>VM erstellen</b>	Kubuntu 24.04 mit <b>VirtIO-GPU</b> .
☐ <b>Wayland aktivieren</b>	Standard in Kubuntu 24.04.
☐ <b>Virtuelle Monitore einrichten</b>	<code>kwin_wayland</code> oder <code>xrandr</code> (XWayland).
☐ <b>NoMachine installieren</b>	Mit <code>EnableWaylandSupport 1</code> .
☐ <b>Firewall anpassen</b>	<code>sudo ufw allow 4000/tcp</code> .
☐ <b>Sicherheit erhöhen</b>	<b>WireGuard-VPN</b> oder <b>Reverse-Proxy</b> .
☐ <b>Testen</b>	Verbindung mit NoMachine-Client (Wayland-Modus).

---

## Fragen?

- Brauchst du eine **detaillierte WireGuard-Anleitung** für Proxmox?
  - Soll ich die **NoMachine-Wayland-Konfiguration** genauer erklären?
  - Oder möchtest du eine **Alternative mit X11** (stabiler, aber ohne Wayland)?
- 

Revision #1

Created 21 September 2025 06:54:35 by Stefan Mechler

Updated 21 September 2025 06:56:17 by Stefan Mechler