

Windows 11 VM + Visual Studio Community on Ubuntu

Step-by-step guide for Ubuntu 24.04 LTS and 26.04 LTS.

KVM + virt-manager + Windows 11 guest + Visual Studio Community.

No custom scripts required. Updated 2026-07-06.

Windows 11 VM + Visual Studio Community on Ubuntu

Step-by-step guide for running Windows 11 in a KVM virtual machine on Ubuntu 24.04 LTS or 26.04 LTS, then installing Visual Studio Community inside the guest.

This guide uses standard Ubuntu packages and virt-manager (GUI). No custom install scripts required.

Why a VM? Visual Studio Community (full IDE) does not run reliably on Wine/Linux. A Windows guest is the supported path for the full IDE, debugger, and designers.

Supported host systems

Host	Display server	KVM / virt-manager
Ubuntu 24.04 LTS	X11 or Wayland (default Wayland on GNOME)	Supported
Ubuntu 26.04 LTS	Wayland only (Xorg session removed)	Supported

Ubuntu 26.04 and X11: Canonical removed the Xorg login session in 26.04, but virt-manager and virt-viewer work on Wayland. X11 client apps still run via XWayland where needed. You do not need an X11 session to create or use KVM VMs.

Desktop environment: Steps below assume GNOME (default Ubuntu). On Kubuntu/Xubuntu/Lubuntu, install the same packages; launch Virtual Machine Manager from the app menu.

What you need

Requirement	Minimum	Recommended
Host OS	Ubuntu 24.04 or 26.04 (64-bit)	Same
CPU	64-bit with Intel VT-x or AMD-V enabled in BIOS	4+ physical cores
Host RAM	16 GiB total (8 GiB free for VM)	32 GiB+ host RAM
Free disk	~90 GiB (ISO + VM disk)	120 GiB+ for VM + VS workloads
VM RAM	8 GiB	16 GiB (Visual Studio is heavy)
VM vCPUs	2	4-6
VM disk	80 GiB	120 GiB qcow2 (sparse; grows as used)
Guest firmware	UEFI + TPM 2.0	Required for Windows 11

Adjust RAM/vCPU down on smaller machines; 8 GiB VM RAM works for light VS workloads but 16 GiB is noticeably smoother.

Part 1 — Enable virtualization (BIOS + Linux)

1.1 BIOS / UEFI

1. Reboot → enter firmware setup (often F2, F10, Del, or Esc — varies by vendor).
2. Enable Intel VT-x, Intel Virtualization Technology, or AMD-V / SVM Mode.
3. Save and reboot.

1.2 Verify on Ubuntu

Open a terminal:

```
# CPU exposes virtualization (one line with vmx or svm is enough)
grep -E 'vmx|svm' /proc/cpuinfo | head -1

# KVM device present
ls -l /dev/kvm
```

- /dev/kvm missing → virtualization disabled in BIOS or blocked by firmware policy.
- Optional check:

```
sudo apt install -y cpu-checker
kvm-ok
```

Expect: KVM acceleration can be used.

1.3 Check free resources

```
free -h
df -h ~
```

Ensure enough RAM and disk per the table above.

Part 2 — Download the Windows 11 ISO

Use a web browser. Microsoft often blocks automated download tools; browser download is the most reliable method.

1. Open Download Windows 11 (<https://www.microsoft.com/en-us/software-download/windows11>).
2. Under Download Windows 11 Disk Image (ISO), select Windows 11 (multi-edition ISO) → Download.
3. Choose language → 64-bit Download.
4. Save the file (typically 5–7 GiB).

Suggested location:

```
mkdir -p ~/VMs/isos
# Move your download to:
~/VMs/isos/win11x64.iso
```

Verify:

```
ls -lh ~/VMs/isos/win11x64.iso
```

Optional integrity check: `sha256sum ~/VMs/isos/win11x64.iso` and compare with Microsoft / rg-adguard listings (<https://files.rg-adguard.net/>).

Part 3 — Install KVM and virt-manager

3.1 Packages

```

sudo apt update
sudo apt install -y \
  qemu-system-x86 \
  qemu-utils \
  libvirt-daemon-system \
  libvirt-clients \
  bridge-utils \
  virt-manager \
  virtinst \
  ovmf \
  swtpm \
  swtpm-tools \
  spice-client-glib-usb-acl-helper \
  acl

```

Package	Purpose
qemu-system-x86	x86_64 emulator
libvirt-daemon-system	VM management service
virt-manager	Graphical VM console
ovmf	UEFI firmware for guest
swtpm	TPM 2.0 emulator (Windows 11 requirement)

3.2 Start libvirt

```

sudo systemctl enable --now libvirtd
sudo systemctl status libvirtd

```

Status should be active (running).

3.3 Allow your user to manage VMs

```

sudo usermod -aG libvirt,kvm "$USER"

```

Log out and log back in (reboot is most reliable on GNOME). Then verify:

```

groups | grep -E 'libvirt|kvm'
virsh --connect qemu:///system list --all

```

The virsh command should run without sudo.

3.4 Enable the default NAT network

```

sudo virsh net-define /usr/share/libvirt/networks/default.xml # skip if already defined
sudo virsh net-autostart default
sudo virsh net-start default
sudo virsh net-list --all

```

default should show active and autostart: yes. This gives the VM internet via NAT (192.168.122.0/24).

Part 4 — Create the virtual machine (virt-manager)

GUI steps are easiest for first-time setup. (CLI alternative: Appendix A (#appendix-a-virt-install-cli).)

4.1 Prepare disk and ISO locations

- ISO: keep at ~/VMs/isos/win11x64.iso (or any path you prefer).
- VM disk: store under libvirt's image directory (avoids home-directory permission issues):

```
sudo mkdir -p /var/lib/libvirt/images
```

If the ISO stays in your home directory, grant read access to libvirt's QEMU user once:

```
sudo setfacl -m u:libvirt-qemu:--x "$HOME"
sudo setfacl -m u:libvirt-qemu:rx "$HOME/VMs" "$HOME/VMs/isos"
sudo setfacl -m u:libvirt-qemu:r "$HOME/VMs/isos/win11x64.iso"
```

(Adjust paths if your ISO lives elsewhere.)

4.2 New VM wizard

1. Launch Virtual Machine Manager (virt-manager) from the application menu.
2. File → New Virtual Machine.
3. Local install media (ISO image or CDRROM) → Forward.
4. Browse → Browse Local → select win11x64.iso.
5. OS type: Microsoft Windows → version Windows 11 → Forward.
6. Memory and CPUs:
 - Memory: 16384 MiB (or 8192 minimum)
 - CPUs: 4-6 (match your host)
7. Storage:
 - Enable storage for this virtual machine
 - Size: 120 GiB (or 80 minimum)
 - Select or create custom storage → Manage → create win11.qcow2 under /var/lib/libvirt/images/
8. Name: win11 → check Customize configuration before install → Finish.

4.3 Required custom settings (before install)

In the configuration window:

Item	Setting
Overview → Firmware	UEFI x86_64 (OVMF_CODE.fd or similar)
CPUs	Copy host CPU configuration optional; host-passthrough if exposed
Add Hardware → TPM	Emulated, version 2.0
SATA Disk 1	Bus SATA for install (simplest; switch to VirtIO later if desired)
NIC	virtio or e1000e (e1000e works without extra drivers during install)
Display	Spice (default)

Click Begin Installation. The Windows installer should start in the viewer window.

Part 5 — Install Windows 11 in the guest

1. Complete language, keyboard, and edition choices.
2. Product key: enter a valid key or I don't have a product key (activate later).
3. Disk: select the virtual disk → Next. If no disk appears, see Troubleshooting (#troubleshooting).
4. Microsoft account vs local account (recent Win11 builds):
 - At "Let's connect you to a network", choose I don't have internet, or temporarily disable the VM NIC in virt-manager.
 - Alternative: Shift+F10 → type OOBE\BYPASSNRO → Enter → reboot → continue offline.
5. Finish personalization; wait for the Windows desktop.

5.1 After Windows is installed

1. Eject install ISO: virt-manager → VM → IDE/CDROM → Disconnect.
2. Windows Update: Settings → Windows Update → install all updates.
3. Activate Windows (optional now): Settings → System → Activation.
4. SPICE guest tools (recommended — clipboard, better display resize):
 - In the VM browser, search for SPICE guest tools or download from your distro's virtio/spice packages page.
 - Install the .exe inside Windows.

5.2 VirtIO drivers (optional, better performance)

For faster disk and network, install Fedora virtio-win drivers (<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso>):

1. virt-manager → Add Hardware → Storage → attach virtio-win.iso as CDROM.
2. In Windows, run virtio-win-gt-x64.exe from the ISO, or install NetKVM and vioscsi drivers manually.
3. Shut down VM → change disk bus to VirtIO and NIC to virtio in virt-manager if you used SATA/e1000e for install.

Part 6 — Install Visual Studio Community

Perform these steps inside the Windows 11 VM.

6.1 Download the installer

1. Open Edge (or another browser) in the guest.
2. Go to Visual Studio Community (<https://visualstudio.microsoft.com/vs/community/>).
3. Click Free download → run VisualStudioSetup.exe.

6.2 Choose workloads

In Visual Studio Installer, select workloads for your project type:

Workload	Use when
.NET desktop development	WinForms, WPF, .NET Framework
Desktop development with C++	C++ / MSVC native
ASP.NET and web development	Web APIs, Blazor
.NET Multi-platform App UI	.NET MAUI (extra mobile SDKs)

For a typical C# desktop setup, enable .NET desktop development. Add Desktop development with C++ if you also need MSVC native tools.

Click Install. Download size is often 10–30+ GiB depending on selections; allow time on first install.

6.3 Sign in and license

- Sign in with a Microsoft account (recommended).
- Community is free for individuals, open-source contributors, and small teams per license terms (<https://visualstudio.microsoft.com/license-terms/>).

6.4 Verify installation

1. Start Visual Studio from the Start menu.
2. Create a new project → Console App (C#) → name VsSmokeTest.
3. Build: Build → Build Solution (or Ctrl+Shift+B).
4. Debug: Debug → Start Debugging (or F5).

Success: app builds and runs in the console window.

Optional PowerShell check:

```
& "${env:ProgramFiles}\Microsoft Visual Studio\2022\Community\Common7\IDE\devenv.exe" /?
```

(Path may be 2022 or 18 depending on VS version; adjust if Installer reports a different year folder.)

6.5 Access host project files (optional)

To open repos that live on Ubuntu:

virt-manager → VM → Add Hardware → Filesystem

Field	Example
Driver	virtiofs
Source path	/home/youruser/projects
Target path	hostprojects (mount tag)

In Windows, map the virtio-fs share (may require virtio-fs guest driver from virtio-win). Alternative: Samba/SMB, git clone inside the guest, or SSH with an editor.

Part 7 — Daily use

Task	Action
Open VM manager	App menu → Virtual Machine Manager
Start VM	Select win11 → Open (play icon)
Graceful shutdown	Inside Windows: Start → Shut down; or virt-manager → Shut Down → Shut down
Force off	Shut Down → Force off (last resort)
Snapshots	virt-manager → View → Snapshots (before major VS upgrades)

Headless start from terminal (optional):

```
virsh --connect qemu:///system start win11
virt-viewer win11
```

Troubleshooting

Symptom	Fix
Could not access KVM kernel module	Enable VT-x/AMD-V in BIOS; <code>sudo usermod -aG kvm \$USER</code> ; reboot
Permission denied connecting to libvirt	Confirm libvirt group: <code>groups</code> ; reboot after <code>usermod</code>
Network not found: default	<code>sudo virsh net-start default</code> (see Part 3.4)
TPM 2.0 / PC doesn't meet Win11 requirements	Add TPM 2.0 (emulated) + UEFI firmware in VM config
No disk in Windows Setup	Use SATA disk during install; load VirtIO drivers or switch bus after <code>virtio-win</code>
No network in guest	<code>sudo virsh net-list --all</code> ; start default network
Slow or tiny VM display	Install SPICE guest tools; use <code>virt-viewer</code> / <code>virt-manager</code> SPICE console
Permission error on disk in ~/	Store qcow2 in <code>/var/lib/libvirt/images/</code>
virt-manager won't start on 26.04	Run from app menu on Wayland session; update: <code>sudo apt install --reinstall virt-manager</code>
VS Installer download fails	Check guest internet (NAT network); retry; use wired host network if Wi-Fi blocks NAT
VS install needs more disk	Expand qcow2 or create larger disk; 120 GiB recommended

Appendix A — virt-install (CLI)

For reproducible scripted creation (same parameters as Part 4):

```
virt-install \
  --connect qemu:///system \
  --name win11 \
  --memory 16384 \
  --vcpus 4 \
  --cpu host-passthrough \
  --disk path="/var/lib/libvirt/images/win11.qcow2",size=120,bus=sata,format=qcow2 \
  --cdrom "$HOME/VMs/isos/win11x64.iso" \
  --os-variant win11 \
  --graphics spice,listen=127.0.0.1 \
  --video virtio \
  --network network=default,model=virtio \
  --boot uefi,menu=on \
  --tpm backend.type=emulator,backend.version=2.0,model=tpm-tis \
  --features smm=on \
  --channel spicevmc \
  --input tablet,bus=usb
```

Then: `virt-viewer win11` or open `virt-manager`.

Rollback (remove VM)

```
virsh --connect qemu:///system destroy win11 2>/dev/null || true  
virsh --connect qemu:///system undefine win11 --remove-all-storage
```

Optional: remove packages with `sudo apt remove virt-manager qemu-system-x86 libvirt-daemon-system`.