Step-by-Step Guide: Running Ollama in Proxmox Debian 12 LXC with Intel iGPU (i5-1240P)

Step-by-Step Guide: Running Ollama in Proxmox Debian 12 LXC with Intel iGPU (i5-1240P)
This guide will show you how to:

1. Enable Intel iGPU passthrough to an LXC container in Proxmox.
2. Set up the latest Intel graphics drivers on both the Proxmox host and the container.
3. Install and use Ollama in the container.

1. Prepare Proxmox Host for Intel iGPU Passthrough

a) Install Essential Packages:

```
apt update && apt install -y intel-opencl-icd
```

b) Enable IOMMU and iGPU Features:
Edit GRUB configuration:

```
nano /etc/default/grub
```

Find the line starting with **GRUB_CMDLINE_LINUX_DEFAULT** and change it to:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on i915.enable_gvt=1"
```

Update GRUB:

```
update-grub
```

c) Load Required Kernel Modules:
Edit /etc/modules

```
nano /etc/modules
```

and add:

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
kvmgt
exngt
vfio-mdev
```

Update initramfs:

```
update-initramfs -u -k all
```

d) Enable GUC for i915 (for best iGPU performance, including newer Intel chips):

```
echo "options i915 enable_guc=3" >> /etc/modprobe.d/i915.conf
```

e) Reboot the **Proxmox Host**

```
reboot
```

f) Check iGPU Availability:
After reboot:

```
lspci -nnv | grep VGA
dmesg | grep -e DMAR -e IOMMU
```

You should see the Intel iGPU and confirmation that **IOMMU is enabled**.

2. Configure Proxmox LXC Container for iGPU Passthrough

a) Stop the LXC Container (if running):

```
pct stop <container_id>
```

b) Edit the Container’s Configuration:

```
nano /etc/pve/lxc/<container_id>.conf
```

Add the following lines:

```
lxc.cgroup2.devices.allow: c 226:0 rwm
lxc.cgroup2.devices.allow: c 226:128 rwm
```

```
lxc.mount.entry: /dev/dri/renderD128 dev/dri/renderD128 none bind,optional,create=file
```

c) Start the Container

```
pct start <container_id>
```

3. Install Latest Intel Graphics Drivers in Debian 12 LXC

Recent Intel drivers for iGPU (including Alder Lake in i5-1240P) are included in the Debian 12 kernel and Mesa packages. Extra steps are only needed if you want bleeding-edge features, but most users do not require them.

a) Update the Container

```
apt update && apt upgrade -y
```

b) Ensure Video Group Membership (for user who'll run Ollama):

```
usermod -aG video <your_username>
usermod -aG render <your_username>
```

c) Test iGPU Access:
Inside the container, run:

```
ls /dev/dri
```

You should see "**renderD128**" (and maybe "**card0**")

Check VAAPI info:

```
apt install vainfo -y
vainfo
```

You should see details about your Intel iGPU.

4. Install Ollama in the LXC Container

a) Install Prerequisites:

```
apt install -y curl
```

b) Install Ollama:

```
curl -fsSL https://ollama.com/install.sh | sh
```

This script sets up the **Ollama binary, user, systemd service, and necessary groups**; Ollama will start running on **localhost:11434**

c) (Optional) Enable API Access from Outside Container:
Edit the service to listen on all IPs:

```
systemctl edit ollama.service
```

Add:

```
[Service]
Environment="OLLAMA_HOST=0.0.0.0"
```

Then restart:

```
systemctl restart ollama.service
```

5. Test Ollama

Example usage:

```
ollama pull llama3:8b
ollama run llama3:8b
```

6. Troubleshooting

- If **/dev/dri** is missing, double-check host config and LXC conf file.
- Ensure container is privileged. Unprivileged LXC passthrough is not generally recommended for iGPU.
- iGPU passthrough is best-supported for media and AI apps in newer Intel chips (11th/12th gen+).
- For advanced iGPU features, stay up-to-date with Proxmox and kernel updates.

You now have Ollama running inside a Debian 12 LXC container on Proxmox with full Intel iGPU (i5-1240P) support and the latest available drivers built into Debian!

Ins0mniA

Revision #3
Created 2025-07-27 00:36:44 EEST by Green
Updated 2025-09-04 02:09:06 EEST by Green