

Proxmox : créer un conteneur IA (Ollama + Open WebUI) à partir d'un template

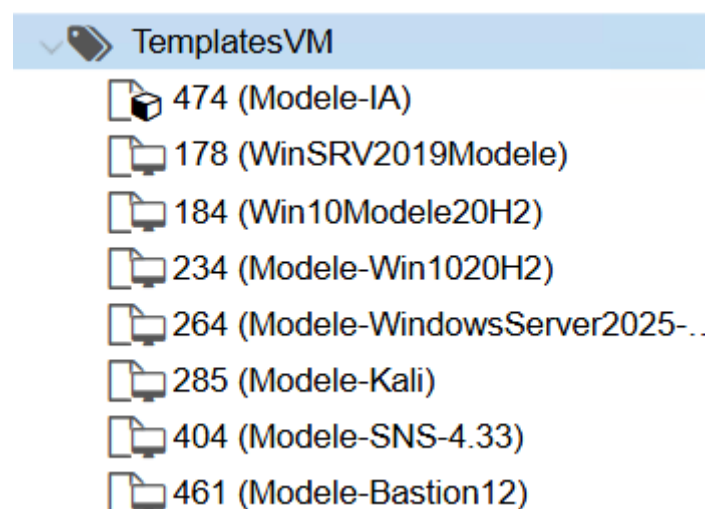
Présentation

Le template proposé permet de disposer d'un conteneur LXC qui lance automatiquement :

- **Ollama** utilisable en ligne de commande ou avec Open WebUI,
- et **Open WebUI** sur l'adresse IP de la VM et les **ports 8080 ou 80**.

Création du conteneur

- Dans le Pool de ressources **TemplatesVM**, cliquez-droit sur le template **474 Modele-IA**



- Renseignez :
 - **IMPORTANT** : le serveur **siohyp2** si vous souhaitez pouvoir utiliser des GPU NVidia (non obligatoire),
 - le nom du conteneur LXC à créer,
 - le pool de ressources du conteneur,
 - **ATTENTION** le mode Full Clone.

Clone CT Template 474

Target node: siohyp2 1

Mode: Full Clone 4

CT ID: 316

Target Storage: Same as source

Hostname: IA 2

Resource Pool: SIOTECHER X 3

Help

Clone

Attendez quelques minutes, le temps de la copie ... de près 40 Gio ...

Caractéristiques de la VM

Ce sont les caractéristiques de départ que vous pourrez modifier, à la hausse ou à la baisse pour certains, en fonction de l'usage du conteneur :

- RAM : 10 Gio,
- Coeurs : 4,
- Disque sur : 100 Gio.

Il a été rajouté au conteneur les **périphériques Passthrough** associés aux deux cartes NVidia Testa T4 (dev0 à dev7). Cela signifie que le conteneur accède directement aux deux cartes Tesla T4 (/dev/nvidia0 et /dev/nvidia1) sans passer par l'hyperviseur Proxmox.

Summary	Add	Edit	Remove	Volume Action	Revert
Resources	Memory	10.00 GiB			
Network	Swap	10.00 GiB			
DNS	Cores	4			
Options	Root Disk	NFS-NAS:474/base-474-disk-0.raw,size=100G			
Task History	Device (dev0)	/dev/nvidia0			
Backup	Device (dev1)	/dev/nvidia1			
Replication	Device (dev2)	/dev/nvidiactl			
Firewall	Device (dev3)	/dev/nvidia-modeset			
Permissions	Device (dev4)	/dev/nvidia-caps/nvidia-cap1			
	Device (dev5)	/dev/nvidia-caps/nvidia-cap2			
	Device (dev6)	/dev/nvidia-uvdm			
	Device (dev7)	/dev/nvidia-uvdm-tools			

Lancement du conteneur

- Lancez le conteneur après la fin du clonage
- Pour ouvrir une session, utilisez le compte **root** avec le mot de passe Sio1234*
- Attendez quelques instant que tous les services soient lancés. La commande **ss -nlt** permet de visualiser les services en écoute afin d'obtenir les ports en écoute (LISTEN) suivants :
 - **80** et **8080** pour **Open WebUI** ;
 - **11434** pour **Ollama**.

root@IA:~# ss -nlt
State Recv-Q Send-Q Local Address:Port Peer Address:Port
LISTEN 0 2048 0.0.0.0:8080 0.0.0.0:*
LISTEN 0 4096 127.0.0.1:2019 0.0.0.0:*
LISTEN 0 100 127.0.0.1:25 0.0.0.0:*
LISTEN 0 4096 127.0.0.1:11434 0.0.0.0:*
LISTEN 0 4096 *:80 *:*
LISTEN 0 4096 *:22 *:*
LISTEN 0 100 [::]:25 [::]:*

Visualiser les ressources consommées par le conteneur

Avec Proxmox, la rubrique **Summary** permet :

- de visualiser la RAM, les coeurs et l'espace disque utilisés en temps réel,

- l'adresse IP du conteneur.

Summary

>_ Console

Resources

Network

DNS

Options

Task History

Backup

Replication

Snapshots

Firewall

Permissions

IA (Uptime: 00:07:23)Debian

Status

running

HA State

none

Node

siohyp2

Unprivileged

Yes

CPU usage

0.02% of 4 CPU(s)

Memory usage

7.83% (801.35 MiB of 10.00 GiB)

SWAP usage

0.00% (0 B of 10.00 GiB)

Bootdisk size

35.07% (34.32 GiB of 97.87 GiB)

IPs

10.10.10.100

fe80::7824:5a93:6d0d:3871

More

- En CLI, la commande **nvidia-smi** permet de visualiser en temps réel, la consommation des ressources des cartes NVidia Tesla T4 :

```
root@IA:~# nvidia-smi
Fri Jan 16 22:09:02 2026

+-----+
| NVIDIA-SMI 590.48.01                Driver Version: 590.48.01      CUDA Version: 13.1     |
+-----+-----+
| GPU   Name                               Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf              Pwr:Usage/Cap |      Memory-Usage | GPU-Util  Compute M. |
|=====  
|  0 Tesla T4                        On          | 00000000:86:00:0 Off |    0MiB / 15360MiB |      0%      Default |
| N/A   44C    P8              13W / 70W    |      0MiB / 15360MiB |          | MIG M. |
+-----+-----+
|  1 Tesla T4                        On          | 00000000:AF:00:0 Off |    0MiB / 15360MiB |      0%      Default |
| N/A   45C    P8              13W / 70W    |      0MiB / 15360MiB |          | MIG M. |
+-----+-----+

+-----+
| Processes:                               |
|  GPU   GI    CI          PID    Type   Process name                      GPU Memory |
|      ID   ID                 |              |           |      Usage |
+-----+-----+
| No running processes found              |
+-----+
```

Accéder à OpenWebUI

- depuis votre ordinateur personnel (Windows, Linux, Mac) lancez votre navigateur pour accéder aux URL suivantes :
 - <http://adresseIPconteneurLXC> (reverse Proxy Caddy utilisé)
 - <http://adresseIPconteneurLXC:8080> (URL par défaut de OpenWebUI lancé avec Python).

Accéder à Ollama depuis un terminal

- dans le terminal du conteneur LXC, créez un compte par exemple **sio** avec un mot de passe de votre choix :

```
adduser sio
```

- depuis votre ordinateur personnel (Windows, Linux, Mac) lancez votre terminal (Powershell, CMD, WSL, terminal linux ou terminal Mac) pour accéder en ssh à Ollama avec la commande suivante en utilisant le compte **sio** et le mot de passe que vous avez défini pour ce compte :

```
ssh sio@adresseIPconteneurLXC
```

- pour passer root utilisez la commande suivant en utilisant le mt de passe du comte root qui est Sio1234* :

```
su -
```

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/reseau/cloud/proxmox/lxccreeria>

Last update: **2026/01/18 18:51**

