CREATION MACHINE VIRTUELLE PROXMOX

**Création de la machine :**

-> Machine : Debian 10

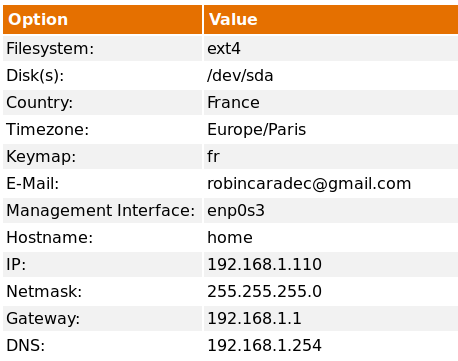
-> Disque virtuel : VMDK 80 GO dynamiquement alloué

**Installation de PMX :**

-> Télécharger l’image ISO de la dernière version de proxmox

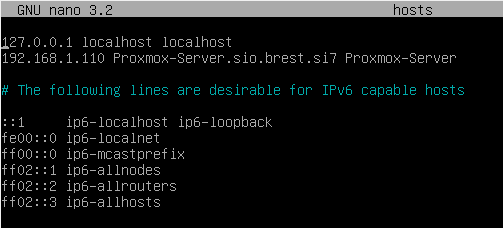
-> L’insérer comme disque dans Stockage

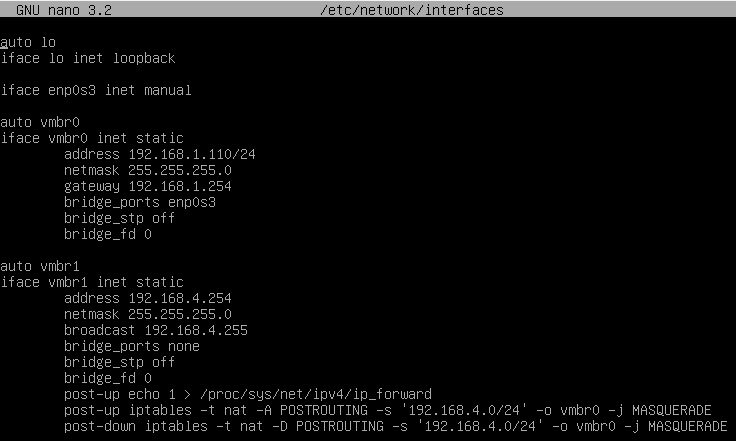
-> Configurer l’installation



Après l’installation : Eteindre la machine et retirer l’image ISO dans Stockage puis lancer la machine.

Pour permettre à la machine d’accéder à internet et que les conteneurs communiquent entre eux :



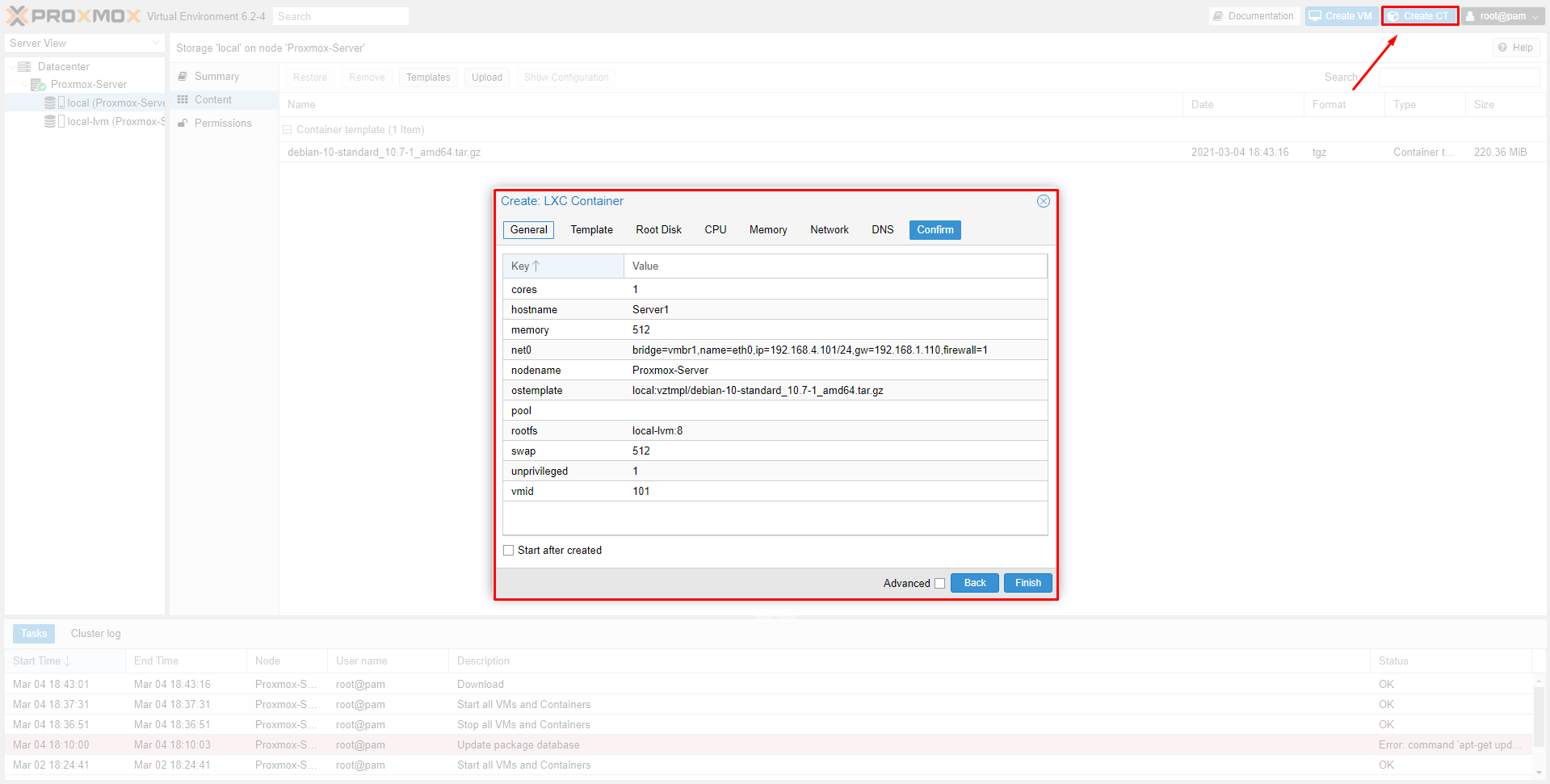


Vmbr1 : Bridge pour créer un LAN permettant aux CTs de communiquer

**ATTENTION :** Le contenu de /etc/hosts doit être en accord avec /etc/hostname

Accéder à l’interface : https://192.168.1.110:8006

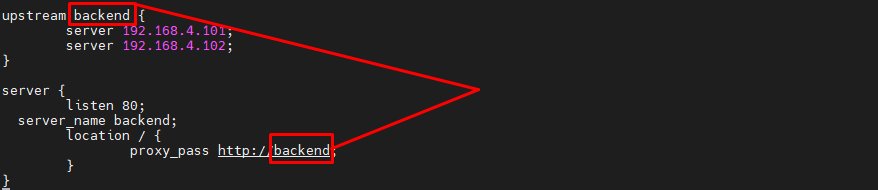
Exemple de création de conteneur :



LOAD BALANCING

1. Installer nginx sur l’host, le container 1 et le container 2
2. Tutoriel : <https://mathisthuault.wordpress.com/2018/09/24/load-balancer-avec-nginx/>

Modifier les fichiers index.html de chaque container pour les différencier

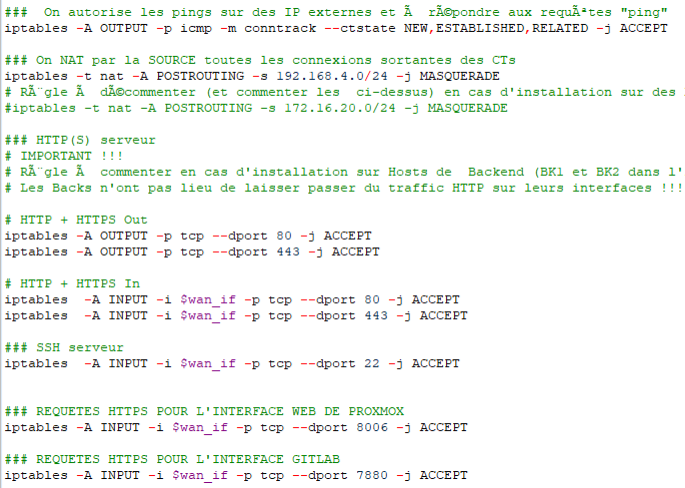
*/etc/nginx/conf.d/load-balancer.conf*

LE FIREWALL

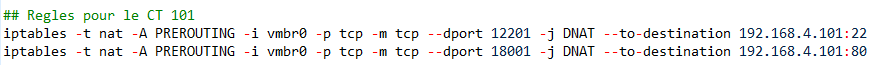
Le but du firewall est d’avoir accès à ses conteneurs à distance à partir d’un port spécifique.

--> Télécharger les fichiers du prof : firewall\_host.sh et MyNatRules.config dans /etc/init.d

firewal\_host2.Sh :



Exemple MyNatRules.config avec le conteneur WEB 101 :

Pour y accéder via http://192.168.1.110:18001

Si on change de réseau : Bien adapter les ips dans firewall\_host

Pour appliquer les changements :

* chmod +x /etc/init.d/firewall\_Host.sh
* /etc/init.d/firewall\_Host.sh
* update-rc.d firewall\_Host.sh defaults (pour lancer à chaque démarrage)

GITLAB

Suivre le tutoriel : <https://www.howtoforge.com/how-to-install-gitlab-with-docker-on-ubuntu-2004/>

ATTENTION : Il faut corriger certains points

* **Ne pas installer containerd au tout début**
* **Ne pas créer de docker-compose.yml, faire plutôt**
* - *docker search gitlab*
* *- docker pull docker.io/gitlab/gitlab-ce*
* *- docker images*
* *- docker run -d --hostname gitlab.sio-brest.si7 \ -p 44443:443 -p 8880:80 -p 2222:22 \ --name gitlab \ --restart always \ --volume /storage/gitlab/config:/etc/gitlab \ --volume /storage/gitlab/logs:/var/log/gitlab \ --volume /storage/gitlab/data:/var/opt/gitlab \ gitlab/gitlab-ce:latest*
* **On peut sauter la partie SSL**

Pour tester : *docker ps* et attendre que le statut passe en healthy puis accéder à http:192.168.1.110:8880

Ne pas oublier d’ajouter le port 8880 (et pas 7880) dans firewall\_host.sh

GitLab ssh : Aller dans l’interface gitlab > edit profile > ssh keys

A FAIRE :

* CT WEB BOUTIQUE
* CT – DNS
* CT SQL Replication
* CT LoadBalancing SQL (HAPROXY)
* CT - Gestion de config (Ansible/Puppet/…)(sur CT)
* CT Supervision

CT DNS : https://www.yoan-jouve.com/mise-en-place-serveur-dns-avec-unbound/ Tu as un CT qui contient Unbound pour communiquer entre tes ct's par nom de machine

CT Gest config : c'est en gros de la gestion de configuration avec Ansible ou Puppet, à partir d'un fichier déployer un Nginx sur un autre ct

CT SQL : il me semble c'est du Load Balancing SQL