

Configuration Raspberry

Important : uniquement sur Raspberry Pi 3 et 4 64 bits.

1. Installation de rocky linux sur la carte USB.

Télécharger l'image à l'adresse <https://rockylinux.org/alternative-images/> l'image de rocky linux, puis la transférer l'image sur le support USB. Je vous conseille d'utiliser d'utiliser le logiciel balenaEtcher (<https://etcher.balena.io/>).

Pour configurer le réseau, le plus simple est d'adapter le fichier ifcfg-eth0 et de le transférer dans le répertoire /etc/sysconfig/network-scripts de la carte USB. Il faut être root pour le copier et mettre les droits 600 a ce fichier.

Si nous utilisons le noyau présent par défaut dans la distribution, nous aurons comme erreur l'absence de hudge page lors du lancement du cluster. Il est nécessaire de recompiler le noyau avec les paramètres suivant sélectionnés (voir <https://unix.stackexchange.com/questions/720878/how-to-enable-hugetlb-controller-in-cgroup-v2-on-ubuntu>):

```
CONFIG_CGROUP_RDMA=y
CONFIG_CGROUP_HUGETLB=y
CONFIG_ARCH_ENABLE_HUGEPAGE_MIGRATION=y
CONFIG_HUGETLBFS=y
CONFIG_HUGETLB_PAGE=y
```

Pour vous faciliter la tâche, je vous fournis le fichier de configuration «.config» pour compiler le noyau. Vous trouverez toutes les explications sur le site :

https://www.raspberrypi.com/documentation/computers/linux_kernel.html#configuring-the-kernel

N'oubliez pas de rajouter

```
make -jXXX
```

Ou XXX est le nombre de cœurs de votre PC. Pour information, sur mon PC i5/8coeurs/32Gram, la compilation du kernel dure 29 minutes.

Appliquez la procédure de compilation croisée, avec transfert des fichiers sur la clef USB.

Il est aussi nécessaire d'ajouter « cgroup_enable=memory » au fichier « /boot/cmdline.txt ».

Une bonne présentation de cgroup se trouve sur le site

<https://downey.io/blog/exploring-cgroups-raspberry-pi>

Une fois les modifications faites, insérer la carte dans le raspberry, et le démarrer.

2. Configuration de rocky linux sur le raspberry.

Un compte existe par défaut : rocky/rockylinux.

Regarder dans le fichier

/var/log/messages

les erreurs éventuelles et vérifier que vous avez bien

```
[root@masterv2 etudiant]# cat /sys/fs/cgroup/cgroup.controllers  
cpuset cpu io memory hugetlb pids rdma
```

Pour rappel, eth0 est down si le raspberry n'est pas connecté au réseau.

Vérifier que kvm-bridge est bien détruit sur l'hôte, sinon les trames sont redirigées vers la passerelle.

```
$ virsh net-list --all  
$ virsh net-destroy bridge-kvm
```

Configurer l'hôte pour être une passerelle

```
$ echo 1 > /proc/sys/net/ipv4/ip_forward  
$ iptables -t nat -A POSTROUTING -o usb0 -j MASQUERADE
```

Et supprimez le firewall du routeur (ici mon pc dell)

Pour utiliser ansible, transférez la clef ssh public avec ssh-copy-id. Ceci n'est pas faisable en copiant directement le fichier durant l'étape 1, car les droits SE ne sont pas bon.

```
$ ssh-copy-id rocky@10.54.56.100
```

Créer l'utilisateur étudiant

```
$ useradd etudiant  
$ passwd etudiant  
$ usermod -aG wheel etudiant
```

Pour permettre à l'étudiant d'élever ses privilèges via sudo sans mot de passe, rajoutez

```
$ echo etudiant ALL=(ALL) NOPASSWD:ALL > /etc/sudoers.d/90-init-users  
$ chmod 440 90-init-users
```

Désactivez firewalld

```
$ systemctl status firewalld.service  
$ systemctl stop firewalld.service  
$ systemctl disable firewalld.service
```

Maintenant, votre raspberry est prêt pour l'installation de k8s.