

INSTALLATION DE L'OS

1. Installer Raspberry Pi Imager
2. Insérer Carte SD
3. Sur le site de ClusterHAT, télécharger les images suivantes afin d'avoir les scripts ClusterCTRL permettant de détecter et contrôler les RP0 :
 - a. CNAT - Desktop Controller -> pour le RP4
 - b. Lite Bookworm image P1 -> pour le RP01
 - c. Lite Bookworm image P2 -> pour le RP02
 - d. Lite Bookworm image P3 -> pour le RP03
 - e. Lite Bookworm image P4 -> pour le RP04
4. Sur Raspberry Pi Imager : Indiquer : Modèle du Raspberry (RPI4 ou RPI Zéro)
5. Indiquer le système d'exploitation précédemment téléchargé correspondant
6. Indiquer le stockage (carte SD)



7. Dans les paramètres de RP Imager, Activer SSH et donner un nom et un mot de passe à l'os (RPI4 et RP01, 02, 03 et 04)
8. Flasher la carte SD
9. Enlever l'adaptateur et l'insérer dans le port micro SD correspondant sur le cluster

RPI 4

1. Brancher le cluster (Alim + Ethernet)
2. Scanner son réseau avec nmap pour récupérer l'IP du RPI4
3. Se connecter en SSH avec `ssh rpi4@192.168.X.X`
4. Mettre à jour avec `sudo apt update && upgrade`
5. Après avoir branché le hub avec le raspberry 4, on peut alimenter et allumer les RP0 avec la commande `clusterctrl on` (off pour les éteindre)

6. On peut ensuite voir que les RP0 sont bien allumés et connectés au sous-réseau du RP4 avec la commande `arp -i br0` qui permet de lister les adresse ip et mac reconnu par l'interface br0 (sous réseau du RP4)
7. On peut donc se connecter en SSH au différent RP0 avec:

```
ssh rp01@172.19.181.1
ssh rp02@172.19.181.2
ssh rp03@172.19.181.3
ssh rp04@172.19.181.4
```

SITE WEB

Pour héberger le site web et sa base de donnée sur le RPI4, nous avons :

1. Installer apache2 avec `sudo apt install apache2`. On peut ensuite vérifier que son installation fonctionne correctement en écrivant l'adresse IP du Raspberry dans un navigateur, on voit qu'une page apache s'affiche.
2. Installer PHP 8.2.29 avec `sudo apt install php libapache2-mod-php` (`libapache2-mod-php` permet à Apache d'interpréter les pages.php)
On vérifie la version avec `php --version`
Pour tester que php est correctement installé, on crée un fichier info.php

```
sudo nano /var/www/html/info.php
```

Dans cette page, on écrit et enregistre la fonction :

```
<?php phpinfo(); ?>
```

En se connectant à info.php sur un navigateur avec

<http://192.168.1.85/info.php>, on voit bien d'info PHP d'afficher

INSTALLATION ET CONFIGURATION DE LA BASE DE DONNÉES

1. Installer MariaDB avec `sudo apt install mariadb-server php-mysql`
2. Pour configurer MariaDB, nous utilisons la commande `sudo mysql_secure_installation`
Pour notre BD, nous avons choisi la configuration suivante

Étape	Réponse	Explication
Switch to unix_socket authentication	n	L'authentification par mot de passe classique est conservée pour l'utilisateur root.
Change the root password	n	Le mot de passe root ayant déjà été défini, il n'est pas nécessaire de le modifier.
Remove anonymous	y	Les comptes anonymes

users		pouvant se connecter à MariaDB sont supprimés pour renforcer la sécurité.
Disallow root login remotely	y	La connexion distante de l'utilisateur root est interdite afin de sécuriser l'accès.
Remove test database and access to it	y	La base de test accessible à tous est supprimée, réduisant les risques potentiels.
Reload privilege tables now	y	Les droits sont rechargés afin que les modifications soient appliquées immédiatement.

3. On peut ensuite se connecter à MariaDB avec `sudo mysql -u root -p`
4. On créer ensuite un user admin

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY '.....';
GRANT ALL PRIVILEGES ON *.* TO 'admin'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

5. On installe phpmyadmin pour pouvoir modifier la BD avec une interface graphique avec `sudo apt install phpmyadmin`
6. Une fois l'installation terminée, on peut y accéder depuis un navigateur avec <http://192.168.1.85/phpmyadmin> avec l'user admin.