

**Récupération de la sauvegarde de la base de données + scripts sauvedb et restauredb, et du DokuWiki**

- On compresse d'abord le fichier avec la commande **tar -cvzf [nom du fichier].tar.tgz**
- Sauvegardes récupérées grâce à la commande **python3 -m http.server 9000**
- Cette commande nous permet d'accéder à la page web de notre serveur et de télécharger les fichiers
- Il n'y a plus qu'à aller sur un navigateur et taper dans la barre de recherche [http://ap23/nom\\_du\\_fichier:9000](http://ap23/nom_du_fichier:9000)
- On télécharge sur notre machine physique les fichiers qui nous intéressent (les deux sauvegardes + données du DokuWiki)
- Il n'y a plus qu'à renvoyer vers le nouveau serveur avec :  
**scp /chemin/vers/le/fichier utilisateur@adresse\_ip\_ou\_nom\_machine\_virtuelle:/chemin/destination/**
- L'ancien serveur peut désormais être supprimé.

**Pour l'installation du nouveau serveur AP23 (conteneur) installé avec Debian 11.7 - mises à jour effectuées - i1-c1 :**

- Nous avons ouvert un navigateur et nous sommes allés à l'URL suivant : <https://pve2:8006> pour se connecter à la machine pve2.
- Nous avons entrés les identifiants et le mot de passe requis pour se connecter.
- Une fois connecté, on a cliqué sur "Créer un conteneur" situé en haut à droite de l'interface pour commencer la création d'un nouveau conteneur.
- Dans la fenêtre de création, on a attribué l'ID CT : 2423 (pour ap23) pour identifier le groupe.
- Nous avons donné un nom au conteneur, ici "ap23-2024", puis cliqué sur "Suivant".
- On a choisi une clé publique d'un des deux postes SISR du groupe pour faciliter les connexions futures, puis cliqué sur "Suivant".

**Pour cela :**

- On a ouvert un terminal sur notre machine locale.
- On a utilisé la commande **ssh-keygen -t rsa** pour générer une nouvelle paire de clés SSH.
- Au moment où nous sommes invité à spécifier le nom du fichier où enregistrer la clé, on a appuyé sur Entrée pour accepter la valeur par défaut (généralement id\_rsa).
- Ensuite, nous étions invité à saisir une phrase de passe. Nous avons alors laissé le champ vide pour créer une clé sans mot de passe. On a appuyé sur Entrée pour continuer.
- La paire de clés SSH (une clé privée et une clé publique) a été générée et enregistrée dans notre répertoire ~/.ssh/.
- On a utilisé la commande **cat ~/.ssh/id\_rsa.pub** pour afficher le contenu de la clé publique.
- On avait plus qu'à copier/coller la clé publique à l'endroit demandé.
- On a ensuite repris la suite du paramétrage du futur conteneur :
  - Dans l'onglet "Modèles", on a sélectionné "Debian 11...." parmi les options disponibles, puis on a cliqué sur "Suivant".
  - Dans l'onglet "Disques", on a choisi une taille de 8 Gib pour le disque, puis on a continué en cliquant sur "Suivant".
  - Dans l'onglet "CPU", on a laissé les paramètres par défaut, puis on a cliqué sur "Suivant".
  - Dans l'onglet "Mémoire", on a choisi une allocation de 1024 Mo (1 Go), puis on a cliqué sur "Suivant".

- Dans l'onglet "Réseau", on a laissé les options par défaut, puis on a cliqué sur "Suivant".
- Enfin, on a cliqué sur "Terminer" pour finaliser la création du conteneur.
- Une fois le conteneur créé, on l'a démarré et on s'est connecté en utilisant le protocole SSH.
- La procédure de connexion sans mot de passe à fonctionné.

#### Nous avons ensuite fait les tâches habituelles :

- **Timezone configurée, configuration NTP synchronisée avec serveur - i1-c2**

- **Pour la configuration du fuseau horaire (timezone) :**

On utilise la commande **timedatectl** puis on verifie la configuration, on a du changer la timezone pour être conforme avec notre zone géographique (Europe/Paris) grâce a la commande **set-timezone Europe/Paris.**

#### **La configuration de la timezone permet aussi de configurer NTP**

Pour les mises à jour : la commande de base : **apt update** ou **upgrade** qui permet de passer de la version Debian 11.6 à 11.7.

Pour l'installation de paquets de base (apache2, php, php-gd, php-mbstring, mariadb-server, php-mysql) : **apt install 'nom du paquet'**

#### **Voici les tests et vérification que le serveur apache2 est opérationnel - i1-c4:**

- Pour cela on utilise la commande **systemctl status apache2.**

On obtient :

```
* apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>
  Active: active (running) since Mon 2023-05-15 08:17:13 CEST; 6min ago
    Docs: https://httpd.apache.org/docs/2.4/
  Process: 9064 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
 Main PID: 9068 (apache2)
   Tasks: 11 (limit: 19115)
  Memory: 40.3M
     CPU: 1.527s
  CGroup: /system.slice/apache2.service
          |-9068 /usr/sbin/apache2 -k start
          |-9069 /usr/sbin/apache2 -k start
          |-9070 /usr/sbin/apache2 -k start
          |-9071 /usr/sbin/apache2 -k start
          |-9076 /usr/sbin/apache2 -k start
          |-9077 /usr/sbin/apache2 -k start
          |-9078 /usr/sbin/apache2 -k start
          |-9083 /usr/sbin/apache2 -k start
          |-9084 /usr/sbin/apache2 -k start
          |-9085 /usr/sbin/apache2 -k start
          `--9086 /usr/sbin/apache2 -k start
```

May 15 08:17:13 ap23-2024 systemd[1]: Starting The Apache HTTP Server...

**Voici les tests et vérification que serveur mariadb est opérationnel - i1-c5:**

- Pour cela on utilise la commande **systemctl status mariadb.**

On obtient :

```
* mariadb.service - MariaDB 10.5.19 database server
Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor prese>
Active: active (running) since Fri 2023-05-12 11:21:01 CEST; 2 days ago
Docs: man:mariadb(8)
      https://mariadb.com/kb/en/library/systemd/
Process: 886 ExecStartPre=/usr/bin/install -m 755 -o mysql -g root -d /var/>
Process: 887 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_STA>
Process: 889 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && V>
Process: 948 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_ST>
Process: 950 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/S>
Main PID: 936 (mariabdd)
Status: "Taking your SQL requests now..."
Tasks: 8 (limit: 19115)
Memory: 60.8M
CPU: 34.616s
CGroup: /system.slice/mariadb.service
        -936 /usr/sbin/mariabdd

May 12 11:21:01 ap23-2024 mariabdd[936]: 2023-05-12 11:21:01 0 [Note] Reading o>
May 12 11:21:01 ap23-2024 mariabdd[936]: 2023-05-12 11:21:01 0 [Note] Added new>
May 12 11:21:01 ap23-2024 mariabdd[936]: 2023-05-12 11:21:01 0 [Note] /usr/sbin>
May 12 11:21:01 ap23-2024 mariabdd[936]: Version: '10.5.19-MariaDB-0+deb11u2' >
May 12 11:21:01 ap23-2024 systemd[1]: Started MariaDB 10.5.19 database server.
```

- **Pour la configurer la connexion SSH par clé publique depuis au moins 2 machines, accès par le nom DNS, adresse IP conforme - i1-c3 :**

**Rappelons nous que nous avons déjà une machine qui peut se connecter en SSH à notre serveur ap23, lors de la création du conteneur.**

**Pour qu'une autre machine puisse se connecter en SSH au serveur sans mot de passe nous avons :**

- Générés une paire de clés SSH sur la machine physique ou on souhaite se connecter avec : **ssh-keygen -t rsa**
- Copiés la clé publique générée (par défaut, id\_rsa.pub) sur la machine physique avec : **cat ~/.ssh/id\_rsa.pub**
- Nous sommes retournés sur le serveur ap23, puis nous avons collés la paire de clé dans : **nano ~/.ssh/authorized\_keys**

**Pour configurer le proxy, utilisez l'adresse suivante : <http://10.121.38.1:8080>.**