

DESCRIPTION D'UNE SITUATION PROFESSIONNELLE

PARCOURS	SISR <input checked="" type="checkbox"/>	SLAM <input type="checkbox"/>
-----------------	---	--------------------------------------

Lieu de réalisation	Coordonnées entreprise ou centre de formation	Logo
Période de réalisation	Du : 13/10/2022	Au : 13/12/2022
Modalité de réalisation	SEUL <input checked="" type="checkbox"/>	EN EQUIPE <input type="checkbox"/>

Intitulé de la mission	Installation zfs
Description du contexte de la mission	Le besoin de l'entreprise était de mettre en place un système de snapshot linux et de le commenter

Contraintes & Résultat	<div style="background-color: #cccccc; padding: 2px; text-align: center; font-size: small;">Ressources fournies / contraintes techniques / Résultats attendu</div> <p>La solution doit utiliser zfs, elle doit pouvoir être lancer sur la machine à chaque extrémité. La sauvegarde doit être tirer et non pousser pour raison de sécurité. De plus les volumes utilisés doivent être chiffrés pour augmenter la sécurité.</p>
Productions associées	<div style="background-color: #cccccc; padding: 2px; text-align: center; font-size: small;">Liste des documents produits et description</div> <p style="text-align: center; font-size: 1.5em;">X</p>

Modalités d'accès aux productions	<div style="background-color: #cccccc; padding: 2px; font-size: small;">Identifiants, mots de passe, URL d'un espace de stockage et présentation de l'organisation du stockage</div> <p style="text-align: center;"> https://rocksurvie.github.io/ https://github.com/rocksurvie/documentsBTS </p> <p>Les documents sont placés dans des répertoires distincts en fonction du lieu de réalisation et sont classés par année pour les missions réalisées en entreprise.</p>
--	--

Description détaillée de la situation professionnelle retenue et des productions réalisées en mettant en évidence la démarche suivie, les méthodes et les techniques utilisées
<p>Installation</p> <p>Part 1 : installer zfs</p> <pre><pre> apt install zfsutils-linux </pre></pre>

Rechercher les disques disponible

```
fdisk -l
```

Création du disque miroir

```
zpool create rpool mirror disk1 disk2
```

création et compression du système de fichier

```
zfs create -o encryption=on -o keyformat=passphrase rpool/backup  
zfs set compression=lz4 rpool/backup
```

Partage de la clé SSH root

Nous avons ici une volonté de permettre l'accès à ce programme sans mot de passe pour éviter qu'il soit écrit noir sur blanc dans un fichier de configuration

Installation du serveur SSH

```
apt install ssh
```

dans le fichier sshd_config

```
nano /etc/ssh/sshd_config
```

Activer ces paramètre

```
PermitRootLogin yes  
PasswordAuthentication yes
```

Redémarrer le service SSH

```
systemctl restart ssh.service
```

générer, copier, tester la clé SSH

```
ssh-keygen -b 4096  
ssh-copy-id root@192.168.122.135  
ssh root@192.168.122.135
```

installation ensuite sanoid

```
sudo apt install sanoid
```

Vérifier le montage du fichier rpool/backup

```
df -h
```

si le fichier n'est pas dans la liste

```
zfs mount rpool/mp3r
```

crée le lien entre les deux serveurs sans fichier python

```
syncoid root@remote:rpool/data/mydataset rpool/backup/mydataset
```

avec le fichier python

nous avons ici une volonté de pouvoir activer ce programme depuis un cron pour pouvoir le périodiser .

paramétrer le sanoid

crée le répertoire sanoid et le fichier sanoid.conf

```
cd /etc  
mkdir sanoid  
touch sanoid.conf
```

entrer ces paramètres dedans

```
[rpool/backup]  
    use_template = production,demo
```

```
    hourly = 12  
    monthly = 1
```

```
[template_demo]  
    daily = 60
```

```
[template_production]  
    frequently = 0  
    hourly = 36  
    daily = 30  
    monthly = 3  
    yearly = 0  
    autosnap = yes  
    autoprune = yes
```

```
[template_backup]  
    autoprune = yes  
    frequently = 0  
    hourly = 30  
    daily = 90  
    monthly = 12  
    yearly = 0
```

autosnap = no

hourly_warn = 2880

hourly_crit = 3600

daily_warn = 48

daily_crit = 60

[template_hotspare]

autoprune = yes

frequently = 0

hourly = 30

daily = 90

monthly = 3

yearly = 0

autosnap = no

hourly_warn = 4h

hourly_crit = 6h

daily_warn = 2d

daily_crit = 4d

[template_scripts]

pre_snapshot_script = /path/to/script.sh

post_snapshot_script = /path/to/script.sh

pruning_script = /path/to/script.sh

script_timeout = 5

[template_ignore]

autoprune = no

autosnap = no

monitor = no

ajouter le fichier backup-syncoïd.py dans le répertoire suivant

cd /usr/local/sbin

lui ajouter les droits d'exécution

chmod +x backup-syncoïd.py

ajouter le fichier JSON dans

cd /etc

on peut maintenant lancer le fichier

backup-syncoid.py

automatisation du lancement backup-syncoid.py avec lancement toute les 15 minutes

en ROOT se rendre dans

cd /etc/cron.d

crée un fichier backup-syncoid

touch backup-syncoid

entrer les lignes suivantes

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin*  
*/15 * * * * root /usr/local/sbin/backup-syncoid.py >/tmp/test2 2>&1
```

pour désactiver le cron ajouter "#" devant la deuxieme ligne

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin*  
#/15 * * * * root /usr/local/sbin/backup-syncoid.py >/tmp/test2 2>&1
```