

Administration Unix

4IIR

TP n° 3 : Configuration des cibles et gestion des services par Systemd

Objectif :

- Comprendre comment fonctionnent les cibles systemd
- Manipuler les commandes les plus utiles des cibles (targets).
- Gérer les services par systemd

Configuration initiale :

- Ce TP est à réaliser avec une station sous Linux RHEL8

Atelier 1 : Bases de systemd et systemctl

1. Vérifier si le système Linux utilise Systemd

Si vous n'êtes pas sûr de savoir si votre distribution Linux utilise systemd, ouvrez une fenêtre Terminal et exécutez la commande suivante :

```
root@hostname# systemctl --version
```

Cela vous montre le numéro de version de systemd sur votre système Linux, si systemd est installé.

2. Vérifier votre gestionnaire de démarrage

Pour vérifier que votre distribution est gérée par systemd, nous pouvons lancer à partir du terminal :

```
root@hostname# ps aux | head -2
```

Si le PID 1 est systemd c'est que votre système fonctionne avec systemd.

3. Vérifier où sont installés les fichiers de systemd et systemctl

```
root@hostname# whereis systemd
root@hostname# whereis systemctl
```

4. Analyser le temps de démarrage de votre système :

La commande ci-dessous donne le temps total nécessaire au démarrage, ainsi que le temps après lequel on arrive à l'écran de connexion.

```
root@hostname# systemd-analyze
```

5. Analyser le temps pris par chaque processus au démarrage.

La commande **system-analyze** utilisée avec l'option **blame** donne la liste des services en cours, triés par ordre de temps nécessaire à leur initialisation (ordre décroissant)

```
root@hostname# systemd-analyze blame
```

6. Analyser la chaîne critique du démarrage

```
root@hostname# systemd-analyze critical-chain
```

Cette commande est utilisée pour afficher la chaîne critique des unités du système au démarrage. Elle fournit des informations sur la séquence d'activation des unités au démarrage du système, en mettant en évidence les dépendances critiques et les unités qui ont un impact significatif sur le temps de démarrage du système.

Cela peut être utile pour diagnostiquer les problèmes de performance liés au démarrage du système et pour identifier les unités qui peuvent nécessiter une optimisation ou une attention particulière afin d'accélérer le processus de démarrage

Atelier 2 : Configurer les cibles systemd

Pour afficher, modifier, ou configurer des cibles Systemd, vous allez utiliser l'utilitaire `systemctl` comme décrit dans le Tableau ci-dessous. Les commandes `runlevel` et `telinit` sont toujours disponibles dans le système et fonctionnent comme prévu, mais ne sont incluses que pour des raisons de compatibilité et doivent être évitées.

Ancienne commande	Nouvelle commande	Description
<code>runlevel</code>	<code>systemctl list-units --type target</code>	Répertorie les unités de cible actuellement chargées
<code>telinit <i>runlevel</i></code>	<code>systemctl isolate name.target</code>	Modifie la cible actuelle.

1. Afficher la cible par défaut

Au démarrage du système, lorsque le contrôle est transféré de `initramfs` à `systemd`, ce dernier tente d'activer la cible par défaut.

Pour déterminer l'unité de cible qui est utilisée par défaut, exécutez la commande suivante :

```
root@hostname# systemctl get-default
```

Cette commande résout le lien symbolique situé sur `/etc/systemd/system/default.target` et affiche le résultat.

2. Afficher la cible actuelle

Pour répertorier toutes les unités de cible actuellement chargées, il faut saisir la commande suivante:

```
root@hostname# systemctl list-units --type target
```

Cette commande affiche le nom complet de chaque unité de cible (UNIT) suivi d'une note indiquant si l'unité a été chargée (LOAD), son état d'activation de haut niveau (ACTIVE) et de bas niveau (SUB), ainsi qu'une courte description (DESCRIPTION).

3. Modifier la cible par défaut

Pour configurer le système de manière à utiliser une unité de cible différente par défaut, saisissez la commande suivante :

```
root@hostname# systemctl set-default name.target
```

Remplacez **name** par le nom de l'unité de cible que vous souhaitez utiliser par défaut (par exemple, multi-user).

Cette commande remplace le fichier /etc/systemd/system/default.target par un lien symbolique pointant vers /usr/lib/systemd/system/**name.target**, où **name** est le nom de l'unité cible que vous souhaitez utiliser.

Exemple: Configurer le système de manière à utiliser l'unité multi-user.target par défaut

```
root@hostname# systemctl set-default multi-user.target
```

Vous aurez l'affichage suivant:

```
root@hostname# systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target→
/lib/systemd/system/multi-user.target.
```

4. Modifier la cible actuelle

Vous pouvez changer la cible actuelle immédiatement avec l'option **isolate** de la commande **systemctl**.

```
root@hostname# systemctl isolate name.target
```

Remplacer **name** par le nom de l'unité de cible que vous souhaitez utiliser par défaut (par exemple, multi-user). Cette commande met en place l'unité de cible nommée **name** et toutes ses unités dépendantes, et arrête immédiatement toutes les autres.

Exemple : Désactiver l'interface utilisateur graphique de la session actuelle (modifier la cible actuelle et activer la cible multi-user.target) :

```
root@hostname# systemctl isolate multi-user.target
root@hostname# systemctl --no-wall rescue
```

5. Lister toutes les unités défaillantes

```
root@hostname# systemctl --failed
```

6. Vérifier si une unité de type service est activée

```
root@hostname# systemctl is-enabled sshd.service
```

7. Vérifiez si un service est en cours d'exécution

```
root@hostname# systemctl status crond.service
```

Atelier 3 : Gestion de services par systemctl

1. Lister tous les services de votre système (activés et désactivés)

```
# systemctl list-unit-files --type=service
```

2. Démarrer, redémarrer, arrêter, recharger le service SSH

```
# systemctl start sshd.service  
# systemctl restart sshd.service  
# systemctl stop sshd.service  
# systemctl reload sshd.service
```

3. Interrogez systemd pour avoir le statut du service sshd

```
systemctl status sshd.service
```

4. Vérifier l'activation d'un service cron

```
# systemctl is-active crond.service
```

5. Activer le démarrage automatique du service cron au démarrage du système

```
# systemctl enable crond.service
```

6. Désactiver le démarrage automatique du service cron au démarrage du système

```
# systemctl disable crond.service
```

7. Masquer ou démasquer un service

```
# systemctl mask httpd.service  
# systemctl unmask httpd.service
```

Lorsqu'un service est masqué, il devient impossible de le démarrer, de l'arrêter ou de le redémarrer via systemd.

8. Envoyer un signal d'arrêt à un service

```
# systemctl kill sshd  
# systemctl status sshd
```

Atelier 4: Création d'un service personnalisé avec systemd

1. Créer un script simple qui affiche un message dans un fichier.

Par exemple, créez un fichier appelé **custom_service.sh** avec le contenu suivant :

```
#!/bin/bash  
  
echo "Bonjour depuis le service personnalisé de systemd" >  
/tmp/custom_service_output.txt
```

2. Créer un fichier de configuration pour votre service.

Par exemple, créez un fichier appelé **custom_service.service** dans le répertoire **/etc/systemd/system/** avec le contenu ci-dessous :

```
# cd /etc/systemd/system/  
  
# touch custom_service.service
```

```
[Unit]  
Description=Service personnalisé de test  
  
[Service]  
ExecStart=/bin/bash /chemin/vers/votre/script/custom_service.sh  
  
[Install]  
WantedBy=multi-user.target
```

Assurez-vous de remplacer **/chemin/vers/votre/script/** par le chemin absolu vers le répertoire contenant votre script.

3. Gestion du service :

Après avoir créé le fichier de service, exécutez les commandes suivantes pour activer et démarrer le service :

```
# systemctl daemon-reload  
# systemctl enable custom_service.service  
# systemctl start custom_service.service
```

4. Vérification du service :

Vérifiez si le service fonctionne en vérifiant le contenu de **/tmp/custom_service_output.txt**.