

Rejoignez les explorateurs, les bâtisseurs et tous ceux qui ont le courage de proposer des solutions nouvelles à des problèmes anciens. Dans le domaine de l'open source, l'innovation dépend entièrement des personnes qui y travaillent.



# Red Hat Training and Certification

## MANUEL D'EXERCICES (ROLE)

Red Hat Enterprise Linux 8.0 RH134

## RED HAT SYSTEM ADMINISTRATION II

Édition 1





# RED HAT SYSTEM ADMINISTRATION II



**Red Hat Enterprise Linux 8.0 RH134**  
**Red Hat System Administration II**  
**Édition 120190531**  
**Date de publication 20190531**

Auteurs: Fiona Allen, Adrian Andrade, Herve Quatremain, Victor Costea,  
Snehangshu Karmakar, Marc Kesler, Saumik Paul  
Éditeur: Philip Sweany, Ralph Rodriguez, David Sacco, Seth Kenlon, Heather  
Charles

Copyright © 2019 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are  
Copyright © 2019 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but  
not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of  
Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat,  
Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details  
contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed please e-mail  
training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, Hibernate, Fedora, the Infinity Logo, and RHCE are  
trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or  
other countries.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/  
service marks of the OpenStack Foundation, in the United States and other countries and are used with the  
OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack  
Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Collaborateurs : Achyut Madhusudan, Rob Locke, Rudolf Kastl, Prashant Rastogi, Heider Souza,  
Michael Phillips, Dallas Spohn

<b>Conventions de la documentation</b>	<b>ix</b>
<b>Introduction</b>	<b>xi</b>
Red Hat System Administration II .....	xi
Organisation de l'environnement de formation .....	xii
Internationalisation .....	xvi
<b>1. Amélioration de la productivité de la ligne de commande</b>	<b>1</b>
Écriture de scripts bash simples .....	2
Exercice guidé: Écriture de scripts bash simples .....	6
Exécution plus efficace des commandes à l'aide de boucles .....	9
Exercice guidé: Exécution plus efficace des commandes à l'aide de boucles .....	15
Correspondance du texte dans les sorties de commande avec des expressions régulières .....	17
Exercice guidé: Correspondance du texte dans les sorties de commande avec des expressions régulières .....	26
Open Lab: Amélioration de la productivité de la ligne de commande .....	29
Résumé .....	35
<b>2. Planification de tâches à venir</b>	<b>37</b>
Planification d'une tâche utilisateur différée .....	38
Exercice guidé: Planification d'une tâche utilisateur différée .....	40
Planification des tâches utilisateur récurrentes .....	44
Exercice guidé: Planification des tâches utilisateur récurrentes .....	47
Planification des tâches système récurrentes .....	50
Exercice guidé: Planification des tâches système récurrentes .....	54
Gestion des fichiers temporaires .....	58
Exercice guidé: Gestion des fichiers temporaires .....	62
Quiz: Planification de tâches à venir .....	66
Résumé .....	70
<b>3. Réglage des performances du système</b>	<b>71</b>
Ajustement des profils de réglage .....	72
Exercice guidé: Ajustement des profils de réglage .....	77
Influence sur l'ordonnancement des processus .....	79
Exercice guidé: Influence sur l'ordonnancement des processus .....	83
Open Lab: Réglage des performances du système .....	87
Résumé .....	93
<b>4. Contrôle de l'accès aux fichiers à l'aide des ACL</b>	<b>95</b>
Interprétation des ACL de fichier .....	96
Quiz: Interprétation des ACL de fichier .....	103
Sécurisation de fichiers à l'aide des ACL .....	106
Exercice guidé: Sécurisation de fichiers à l'aide des ACL .....	111
Open Lab: Contrôle de l'accès aux fichiers à l'aide des ACL .....	116
Résumé .....	126
<b>5. Gestion de la sécurité avec SELinux</b>	<b>127</b>
Modification du mode d'exécution SELinux .....	128
Exercice guidé: Modification du mode d'exécution SELinux .....	132
Contrôle des contextes de fichiers SELinux .....	135
Exercice guidé: Contrôle des contextes de fichiers SELinux .....	139
Ajustement de la politique SELinux avec des valeurs booléennes .....	142
Exercice guidé: Ajustement de la politique SELinux avec des valeurs booléennes .....	144
Analyse et résolution des problèmes liés à SELinux .....	147
Exercice guidé: Analyse et résolution des problèmes liés à SELinux .....	152
Open Lab: Gestion de la sécurité avec SELinux .....	156
Résumé .....	162

<b>6. Gestion du stockage de base</b>	<b>163</b>
Ajout de partitions, de systèmes de fichiers et de montages persistants .....	164
Exercice guidé: Ajout de partitions, de systèmes de fichiers et de montages persistants .....	175
Gestion de l'espace d'échange .....	179
Exercice guidé: Gestion de l'espace d'échange .....	184
Open Lab: Gestion du stockage de base .....	188
Résumé .....	196
<b>7. Gestion des volumes logiques</b>	<b>197</b>
Création de volumes logiques .....	198
Exercice guidé: Création de volumes logiques .....	206
Extension des volumes logiques .....	211
Exercice guidé: Extension des volumes logiques .....	217
Open Lab: Gestion des volumes logiques .....	221
Résumé .....	227
<b>8. Mise en œuvre de fonctionnalités de stockage avancées</b>	<b>229</b>
Gestion du stockage en couches avec Stratis .....	230
Exercice guidé: Gestion du stockage en couches avec Stratis .....	236
Compression et déduplication du stockage avec VDO .....	242
Exercice guidé: Compression et déduplication du stockage avec VDO .....	245
Open Lab: Mise en œuvre de fonctionnalités de stockage avancées .....	249
Résumé .....	258
<b>9. Accès au stockage rattaché au réseau</b>	<b>259</b>
Montage du stockage rattaché au réseau avec NFS .....	260
Exercice guidé: Gestion du stockage rattaché au réseau avec NFS .....	265
Montage automatique du stockage rattaché au réseau .....	269
Exercice guidé: Montage automatique du stockage rattaché au réseau .....	273
Open Lab: Accès au stockage rattaché au réseau .....	279
Résumé .....	286
<b>10. Contrôle du processus de démarrage</b>	<b>287</b>
Sélection de la cible de démarrage .....	288
Exercice guidé: Sélection de la cible de démarrage .....	293
Réinitialisation du mot de passe root .....	296
Exercice guidé: Réinitialisation du mot de passe root .....	300
Correction des problèmes de système de fichiers au démarrage .....	302
Exercice guidé: Correction des problèmes de système de fichiers au démarrage .....	304
Open Lab: Contrôle du processus de démarrage .....	307
Résumé .....	313
<b>11. Gestion de la sécurité réseau</b>	<b>315</b>
Gestion de pare-feu serveur .....	316
Exercice guidé: Gestion de pare-feu serveur .....	324
Contrôle de l'étiquetage de ports SELinux .....	328
Exercice guidé: Contrôle de l'étiquetage de ports SELinux .....	331
Open Lab: Gestion de la sécurité réseau .....	335
Résumé .....	343
<b>12. Installation de Red Hat Enterprise Linux</b>	<b>345</b>
Installation de Red Hat Enterprise Linux .....	346
Exercice guidé: Installation de Red Hat Enterprise Linux .....	351
Automatisation de l'installation avec Kickstart .....	354
Exercice guidé: Automatisation de l'installation avec Kickstart .....	363
Installation et configuration des machines virtuelles .....	366
Quiz: Installation et configuration des machines virtuelles .....	371

Open Lab: Installation de Red Hat Enterprise Linux .....	373
Résumé .....	379
<b>13. Révision complète</b>	<b>381</b>
Révision complète .....	382
Open Lab: Correction des problèmes de démarrage et maintenance des serveurs .....	385
Open Lab: Configuration et gestion des systèmes de fichiers et du stockage .....	392
Open Lab: Configuration et gestion de la sécurité du serveur .....	399



# CONVENTIONS DE LA DOCUMENTATION



## RÉFÉRENCES

Les « références » indiquent où trouver de la documentation externe se rapportant à un sujet.



## NOTE

Une « remarque » est un conseil, un raccourci ou une approche alternative pour la tâche considérée. Le fait d'ignorer une remarque ne devrait pas entraîner de conséquences négatives, mais vous pourriez passer à côté d'une astuce qui vous simplifierait la vie.



## IMPORTANT

Les cadres « Important » détaillent des éléments qui pourraient aisément être négligés : des changements de configuration qui ne s'appliquent qu'à la session en cours ou des services qui doivent être redémarrés pour qu'une mise à jour soit appliquée. Ignorer un cadre « Important » ne vous fera perdre aucune donnée, mais cela pourrait être source de frustration et d'irritation.



## MISE EN GARDE

Un « avertissement » ne doit pas être ignoré. Le fait d'ignorer un avertissement risque fortement d'entraîner une perte de données.



# INTRODUCTION

## RED HAT SYSTEM ADMINISTRATION II

Ce cours est spécialement conçu pour les stagiaires ayant suivi la formation Red Hat System Administration I (RH124). Red Hat System Administration II (RH134) se concentre sur les tâches clés requises pour devenir un administrateur Linux à plein temps et pour valider ces compétences par l'examen Administrateur système certifié Red Hat. Ce cours va plus loin dans l'administration d'Enterprise Linux, en couvrant les systèmes de fichiers et le partitionnement, les volumes logiques, SELinux, les pare-feu et la résolution de problèmes.

### OBJECTIFS DU COURS

- Étendre et approfondir les compétences acquises pendant le cours Red Hat System Administration I (RH124).
- Acquérir les compétences que requiert un administrateur système Red Hat Enterprise Linux certifié RHCSA.

### PUBLIC

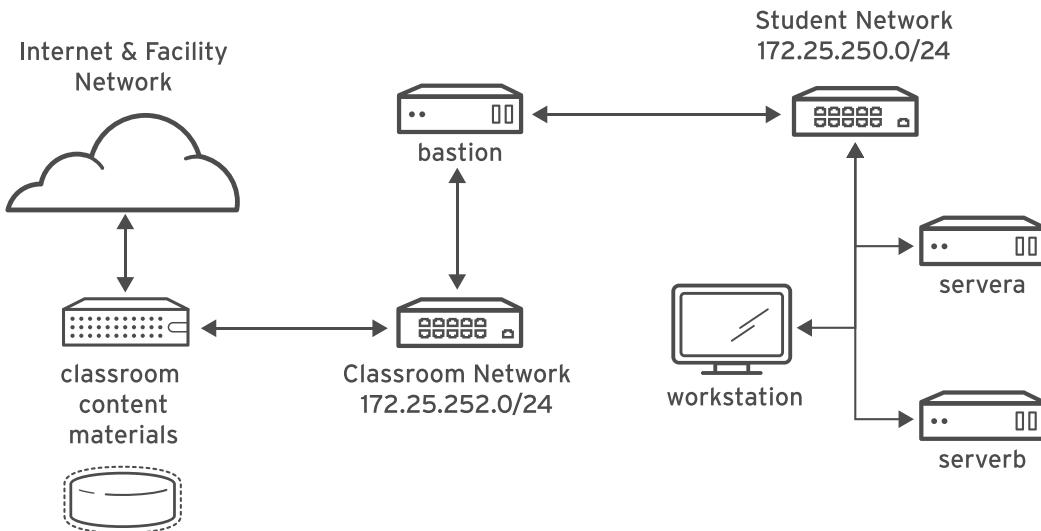
- Ce cours est spécialement conçu pour les stagiaires ayant suivi la formation Red Hat System Administration I (RH124). L'organisation des rubriques fait que le cours RH134 n'est pas approprié comme formation de premier niveau. Les stagiaires qui n'ont suivi aucun cours Red Hat auparavant sont invités à suivre soit System Administration I (RH124) s'ils découvrent Linux, soit RHCSA Fast Track (cours accéléré pour devenir RHCSA, RH200) s'ils ont déjà de l'expérience de l'administration Linux Entreprise.

### CONDITIONS PRÉALABLES

- Le stagiaire doit avoir suivi le cours Red Hat System Administration I (RH124), ou posséder des connaissances équivalentes.

# ORGANISATION DE L'ENVIRONNEMENT DE FORMATION

---



**Figure 0.1: Environnement de formation**

Dans ce cours, le système informatique principal utilisé pour les travaux pratiques est **workstation**. Deux autres machines sont également utilisées par les stagiaires pour ces activités : **servera** et **serverb**. Ces trois systèmes se trouvent dans le domaine DNS `lab.example.com`.

Tous les systèmes informatiques des stagiaires possèdent un compte d'utilisateur standard, **student**, protégé par le mot de passe **student**. Le mot de passe **root** est **redhat** sur tous les systèmes des stagiaires.

## Machines de la salle de classe

NOM DE LA MACHINE	ADRESSES IP	RÔLE
bastion.lab.example.com	172.25.250.254	Système passerelle pour connecter le réseau privé des stagiaires au serveur de la salle de classe (doit toujours être en cours d'exécution)
workstation.lab.example.com	172.25.250.9	Poste de travail graphique utilisé pour l'administration du système
servera.lab.example.com	172.25.250.10	Premier serveur
serverb.lab.example.com	172.25.250.11	Second serveur

La fonction principale de **bastion** est de servir de routeur entre le réseau sur lequel sont connectées les machines des stagiaires et le réseau de la salle de classe. Si le poste de travail

## Introduction

**bastion** est arrêté, les autres machines de stagiaires peuvent uniquement accéder aux systèmes qui se trouvent sur le réseau des stagiaires.

Plusieurs systèmes dans la salle de classe proposent des services d'assistance. Deux serveurs, `content.example.com` et `materials.example.com`, hébergent les logiciels et les supports d'atelier utilisés pour les activités pratiques. Les informations relatives à l'utilisation de ces serveurs sont fournies dans les instructions de ces activités, notamment par la machine virtuelle `classroom.example.com`. Les machines `classroom` et `bastion` doivent toujours être en cours d'exécution pour une utilisation correcte de l'environnement d'atelier.

**NOTE**

Lorsque vous vous connectez à `servera` ou `serverb`, il se peut que vous voyiez un message concernant l'activation de `cockpit`. Le message peut être ignoré.

```
[student@workstation ~]$ ssh student@serverb
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of
known hosts.
Activate the web console with: systemctl enable --now cockpit.socket

[student@serverb ~]$
```

## CONTRÔLE DE VOS SYSTÈMES

Les stagiaires se voient attribuer des ordinateurs distants dans une salle de classe de formation en ligne Red Hat. L'accès s'effectue par le biais d'une application Web hébergée à l'adresse suivante : `rol.redhat.com` [`http://rol.redhat.com`]. Pour se connecter à ce site, les stagiaires doivent utiliser leurs informations d'identification du Portail client Red Hat.

### Contrôle des machines virtuelles

Les machines virtuelles de votre environnement de formation sont contrôlées sur une Page Web. L'état de chaque machine virtuelle de la salle de classe est affiché sur la page de l'onglet Online Lab.

#### États de la machine

ÉTAT DE LA MACHINE VIRTUELLE	DESCRIPTION
STARTING	La machine virtuelle est en cours de démarrage.
STARTED	La machine virtuelle est en cours d'exécution et disponible (ou, pendant le démarrage, le sera bientôt.)
STOPPING	La machine virtuelle est en cours d'arrêt.
STOPPED	La machine virtuelle est complètement arrêtée. Au démarrage, la machine virtuelle affiche le même état que lors de son arrêt (le disque est préservé).
PUBLISHING	La création initiale de la machine virtuelle est en cours.

ÉTAT DE LA MACHINE VIRTUELLE	DESCRIPTION
WAITING_TO_START	La machine virtuelle est en attente du démarrage d'autres machines virtuelles.

Selon l'état de la machine, une sélection des actions suivantes est disponible.

#### Actions de machine/salle de classe

BOUTON OU ACTION	DESCRIPTION
PROVISION LAB	Permet de créer la salle de classe ROL. Crée toutes les machines virtuelles nécessaires pour la salle de classe et les démarre. Cette procédure peut prendre plusieurs minutes.
DELETE LAB	Permet de supprimer la salle de classe ROL. Détruit toutes les machines virtuelles de la salle de classe. <b>Attention : tous les travaux générés sur les disques seront perdus.</b>
START LAB	Permet de démarrer toutes les machines virtuelles de la salle de classe.
SHUTDOWN LAB	Permet d'arrêter toutes les machines virtuelles de la salle de classe.
OPEN CONSOLE	Permet d'ouvrir un nouvel onglet dans le navigateur et de se connecter à la console de la machine virtuelle. Les stagiaires peuvent se connecter directement à la machine virtuelle et exécuter des commandes. Dans la plupart des cas, les stagiaires doivent se connecter à la machine virtuelle de la station de travail et utiliser <b>ssh</b> pour se connecter aux autres machines virtuelles.
ACTION → Start	Permet de démarrer (allumer) la machine virtuelle.
ACTION → Shutdown	Permet d'éteindre correctement la machine virtuelle, en conservant le contenu sur son disque.
ACTION → Power Off	Force l'arrêt de la machine virtuelle, en conservant le contenu du disque. Cela équivaut à couper l'alimentation d'une machine physique.
ACTION → Reset	Force l'arrêt de la machine virtuelle et réinitialise le disque à son état initial. <b>Attention : tous les travaux générés sur le disque seront perdus.</b>

Au début d'un exercice, si vous êtes invité à réinitialiser un seul nœud de machine virtuelle, cliquez sur ACTION → Reset pour la machine virtuelle concernée.

Au début d'un exercice, si vous êtes invité à réinitialiser l'ensemble des machines virtuelles, cliquez sur ACTION → Reset

Si vous souhaitez que l'environnement de formation retourne à son état d'origine du début du cours, vous pouvez cliquer sur DELETE LAB pour supprimer l'ensemble de l'environnement de

formation. Une fois l'exercice pratique supprimé, vous pouvez cliquer sur PROVISION LAB pour déployer un nouvel ensemble de systèmes de salle de classe.



### MISE EN GARDE

L'opération DELETE LAB ne peut pas être annulée. Tous les travaux que vous aurez terminés jusqu'ici dans l'environnement de formation seront perdus.

## Minuterie d'arrêt automatique

L'inscription à la formation en ligne Red Hat (ROL) confère aux stagiaires le droit d'utiliser l'ordinateur pendant un temps donné. Afin de les aider à conserver le temps d'utilisation de l'ordinateur qui leur est alloué, une minuterie d'arrêt automatique est associée à la salle de classe ROL. Celle-ci ferme l'environnement de formation à l'expiration du temps prévu.

Pour régler la minuterie, cliquez sur MODIFY afin que la boîte de dialogue New Autostop Time s'affiche. Définissez le nombre d'heures jusqu'à l'arrêt automatique de la classe. Cliquez sur ADJUST TIME pour appliquer cette modification aux paramètres de la minuterie.

# INTERNATIONALISATION

## SÉLECTION DE LA LANGUE PAR UTILISATEUR

Il se peut que vos utilisateurs veuillent utiliser, pour leur environnement de bureau, une langue différente de celle utilisée par l'ensemble du système. Il se peut aussi qu'ils veuillent utiliser une autre disposition de clavier ou une autre méthode de saisie pour leur compte.

### Paramètres linguistiques

Dans l'environnement de bureau GNOME, l'utilisateur peut être invité, lors de sa première connexion, à configurer la langue et la méthode de son choix. Dans le cas contraire, la manière la plus simple pour un utilisateur d'ajuster les réglages de langue et de méthode de saisie est d'utiliser l'application Region & Language.

Vous pouvez démarrer cette application de deux manières. Vous pouvez exécuter la commande **gnome-control-center region** depuis la fenêtre de terminal ou sur la barre du haut, à partir du menu système situé dans le coin droit, sélectionnez le bouton des paramètres (dont l'icône ressemble à un tournevis croisé et une clé) en bas à gauche du menu.

Dans la fenêtre qui s'ouvre, sélectionnez Region & Language. Cliquez sur la case Language et sélectionnez la langue souhaitée dans la liste qui s'affiche. Cela met également à jour le réglage Formats pour qu'il corresponde aux réglages par défaut pour cette langue. Ces modifications entreront en vigueur la prochaine fois que vous vous connectez.

Ces paramètres affectent l'environnement de bureau GNOME et toutes les applications qui y sont lancées, telles que **gnome-terminal**. Toutefois, par défaut, ils ne s'appliquent pas à ce compte si l'accès a été réalisé via une connexion **ssh** à partir d'un système distant ou d'une connexion texte sur une console virtuelle (ex. : **tty5**).



#### NOTE

Vous pouvez faire en sorte que votre environnement de shell utilise le même paramètre **LANG** que votre environnement graphique, même lorsque vous vous connectez par l'intermédiaire d'une console virtuelle en mode texte ou par **ssh**. Pour ce faire, vous pouvez placer le code suivant ou son équivalent dans votre fichier **~/.bashrc**. Ce code fourni en exemple règle la langue utilisée pour une connexion en mode texte pour qu'elle corresponde à celle configurée pour l'environnement de bureau GNOME :

```
i=$(grep 'Language=' /var/lib/AccountsService/users/${USER} \
    | sed 's/Language=/"/')
if [ "$i" != "" ]; then
    export LANG=$i
fi
```

Le japonais, le coréen, le chinois et d'autres langues à jeu de caractères autre que le latin peuvent ne pas s'afficher correctement sur les consoles virtuelles en mode texte.

On peut obliger chaque commande à utiliser une autre langue, en réglant la variable LANG depuis la ligne de commande :

```
[user@host ~]$ LANG=fr_FR.utf8 date  
jeu. avril 25 17:55:01 CET 2019
```

Les commandes suivantes continuent d'utiliser la langue par défaut du système. La commande **locale** peut être utilisée pour déterminer la valeur actuelle de LANG, ainsi que d'autres variables d'environnement connexes.

## Paramètres de la méthode de saisie

Dans Red Hat Enterprise Linux 7 ou version ultérieure, GNOME 3 utilise automatiquement le système de sélection de méthode de saisie IBus qui permet de changer facilement et rapidement la disposition du clavier et les méthodes de saisie.

L'application Region & Language peut aussi servir à activer d'autres méthodes de saisie. Dans la fenêtre de l'application Region & Language, le cadre Input Sources présente les méthodes de saisie actuellement disponibles. Par défaut, English (US) peut être la seule méthode disponible. Sélectionnez English (US), puis cliquez sur l'icône du clavier pour afficher la disposition actuelle du clavier.

Pour ajouter une nouvelle méthode de saisie, cliquez sur le bouton + dans le coin inférieur gauche de la fenêtre Input Sources. Une fenêtre Add an Input Source s'ouvre. Sélectionnez votre langue, puis la méthode de saisie ou la disposition de clavier souhaitée.

Lorsque plusieurs méthodes de saisie ont été configurées. L'utilisateur peut passer rapidement de l'une à l'autre en saisissant **Super+Space** (parfois appelé **Windows+Space**). Par ailleurs, un *indicateur d'état* s'affiche dans la barre supérieure de l'environnement GNOME. Celui-ci a deux fonctions : il indique la méthode de saisie active et joue le rôle de menu vous permettant de passer d'une méthode de saisie à l'autre ou de sélectionner les fonctions avancées de méthodes de saisie plus complexes.

Certaines des méthodes sont marquées par des engrenages, qui indiquent qu'elles ont des options de configuration et des possibilités avancées. Par exemple, la méthode de saisie japonaise Japonais (Kana Kanji) permet à l'utilisateur de préparer un texte en caractères latins et d'utiliser les touches **Flèche vers le bas** et **Flèche vers le haut** pour sélectionner les caractères à utiliser.

Les anglophones américains peuvent également trouver cette méthode utile. Pour English (United States) par exemple, la disposition de clavier est English (international avec touches mortes en AltGr), qui considère la touche **AltGr** (ou la touche **Alt**) de droite) sur un clavier PC à 104-105 touches comme une touche de modification « Maj secondaire » et comme touche d'activation des touches mortes pour la saisie des caractères supplémentaires. Le Dvorak et d'autres dispositions sont également proposées.



### NOTE

Si vous connaissez le point de code Unicode du caractère, vous pouvez le saisir dans l'environnement de bureau GNOME. Appuyez sur **Ctrl+Maj+U**, suivi du point de code. Après avoir appuyé sur les touches **Ctrl+Maj+U**, un **u** souligné s'affiche pour indiquer que le système attend la saisie du code du caractère.

Par exemple, la lettre minuscule grecque lambda a pour point de code U+03BB et peut être saisie en appuyant sur **Ctrl+Maj+U**, puis **03BB**, et ensuite **Entrée**.

## PARAMÈTRES LINGUISTIQUES PAR DÉFAUT POUR L'ENSEMBLE DU SYSTÈME

La langue par défaut du système est configurée sur US English (Anglais des États-Unis), avec le jeu de caractères Unicode UTF-8 (**en\_US.utf8**), mais cela peut être changé pendant ou après l'installation.

Depuis la ligne de commande, l'utilisateur **root** peut modifier les paramètres linguistiques à l'échelle du système à l'aide de la commande **localectl**. Si la commande **localectl** est exécutée sans argument, elle affiche les paramètres linguistiques à l'échelle du système.

Pour définir une langue par défaut au niveau du système, exécutez la commande **localectl set-locale LANG=locale**, où *locale* est la valeur appropriée pour la variable d'environnement LANG correspondante décrite dans le tableau « Référence des codes de langue » du présent chapitre. Les changements seront pris en compte lors de la prochaine connexion de l'utilisateur et seront stockés dans le fichier **/etc/locale.conf**.

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

Dans GNOME, les administrateurs peuvent modifier ce paramètre dans Region & Language en cliquant sur le bouton Login Screen dans le coin supérieur droit de la fenêtre. La modification de la langue (Language) de l'écran de connexion graphique ajustera également le paramètre linguistique pour l'ensemble du système, stocké dans le fichier de configuration **/etc/locale.conf**.



### IMPORTANT

Les consoles virtuelles en mode texte telles que **tty4** sont plus limitées en ce qui concerne les polices qu'elles peuvent afficher que les terminaux d'une console virtuelle fonctionnant sous un environnement graphique, ou les pseudoterminal pour les sessions **ssh**. Par exemple, les caractères japonais, coréens et chinois peuvent ne pas s'afficher correctement dans une console virtuelle en mode texte. Pour cette raison, vous devriez envisager d'utiliser l'anglais ou une autre langue avec un jeu de caractères latins pour la langue par défaut pour l'ensemble du système.

De même, les consoles virtuelles en mode texte reconnaissent moins de méthodes de saisie. Ce point est géré séparément de l'environnement graphique du bureau. On peut configurer les paramètres de saisie globaux par l'intermédiaire de **localectl**, à la fois pour les consoles virtuelles en mode texte et pour l'environnement graphique. Voir les pages du manuel **localectl(1)** et **vconsole.conf(5)** pour plus d'informations.

## MODULES LINGUISTIQUES

Des paquetages RPM spéciaux appelés *langpacks* installent des paquetages de langue qui prennent en charge des langues spécifiques. Ces langpacks utilisent des dépendances pour installer automatiquement des paquetages RPM supplémentaires contenant des localisations, des dictionnaires et des traductions pour les autres paquetages logiciels de votre système.

Pour lister les langpacks installés et susceptibles d'être installés, utilisez **yum list langpacks-\* :**

```
[root@host ~]# yum list langpacks-*  
Updating Subscription Management repositories.  
Updating Subscription Management repositories.  
Installed Packages  
langpacks-en.noarch      1.0-12.el8        @AppStream  
Available Packages  
langpacks-af.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms  
langpacks-am.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms  
langpacks-ar.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms  
langpacks-as.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms  
langpacks-ast.noarch      1.0-12.el8        rhel-8-for-x86_64-appstream-rpms  
...output omitted...
```

Pour ajouter une prise en charge linguistique, installez le paquetage langpacks approprié. Par exemple, la commande suivante ajoute la prise en charge du français :

```
[root@host ~]# yum install langpacks-fr
```

Utilisez **yum repoquery --what supplements** pour déterminer quels paquetages RPM peuvent être installés par un langpack :

```
[root@host ~]# yum repoquery --what supplements langpacks-fr  
Updating Subscription Management repositories.  
Updating Subscription Management repositories.  
Last metadata expiration check: 0:01:33 ago on Wed 06 Feb 2019 10:47:24 AM CST.  
glibc-langpack-fr-0:2.28-18.el8.x86_64  
gnome-getting-started-docs-fr-0:3.28.2-1.el8.noarch  
hunspell-fr-0:6.2-1.el8.noarch  
hyphen-fr-0:3.0-1.el8.noarch  
libreoffice-langpack-fr-1:6.0.6.1-9.el8.x86_64  
man-pages-fr-0:3.70-16.el8.noarch  
mythes-fr-0:2.3-10.el8.noarch
```



### IMPORTANT

Les paquetages langpacks utilisent les *dépendances faibles* RPM afin d'installer des paquetages supplémentaires uniquement lorsque le paquetage principal qui en a besoin est également installé.

Par exemple, lors de l'installation de *langpacks-fr* comme le montrent les exemples précédents, le paquetage *mythes-fr* ne sera installé que si le dictionnaire des synonymes *mythes* est également installé sur le système.

Si *mythes* est ensuite installé sur ce système, le paquetage *mythes-fr* sera également automatiquement installé en raison de la faible dépendance du paquetage *langpacks-fr* déjà installé.



## RÉFÉRENCES

Pages du manuel **locale(7)**, **localectl(1)**, **locale.conf(5)**, **vconsole.conf(5)**, **unicode(7)** et **utf-8(7)**

Les conversions entre le nom des présentations X11 de l'environnement graphique de bureau et leur nom dans **localectl** se trouvent dans le fichier **/usr/share/X11/xkb/rules/base.lst**.

# RÉFÉRENCE DES CODES DE LANGUE



## NOTE

Ce tableau peut ne pas refléter tous les langpacks disponibles sur votre système. Utilisez **yum info langpacks-SUFFIX** pour obtenir plus d'informations sur un paquetage particulier de langpacks.

### Codes de langue

LANGUE	SUFFIXE LANGPACKS	VALEUR \$LANG
Anglais (États-Unis)	en	en_US.utf8
Assamais	comme	as_IN.utf8
Bengali	bn	bn_IN.utf8
Chinois (simplifié)	zh_CN	zh_CN.utf8
Chinois (traditionnel)	zh_TW	zh_TW.utf8
Français	FR	fr_FR.utf8
Allemand	de	de_DE.utf8
Gujarati	gu	gu_IN.utf8
Hindi	hi	hi_IN.utf8
Italien	it	it_IT.utf8
Japonais	ja	ja_JP.utf8
Kannada	kn	kn_IN.utf8
Coréen	ko	ko_KR.utf8
Malayalam	ml	ml_IN.utf8
Marathi	mr	mr_IN.utf8
Odia	ou	or_IN.utf8
Portugais (brésilien)	pt_BR	pt_BR.utf8

<b>LANGUE</b>	<b>SUFFIXE LANGPACKS</b>	<b>VALEUR \$LANG</b>
Pendjabi	pa	pa_IN.utf8
Russe	ru	ru_RU.utf8
Espagnol	es	es_ES.utf8
Tamoul	ta	ta_IN.utf8
Télougou	te	te_IN.utf8



## CHAPITRE 1

# AMÉLIORATION DE LA PRODUCTIVITÉ DE LA LIGNE DE COMMANDE

### PROJET

Exécuter les commandes plus efficacement en utilisant les fonctionnalités avancées du shell bash, des scripts shell et divers utilitaires fournis par Red Hat Enterprise Linux.

### OBJECTIFS

- Automatiser des séquences de commandes en écrivant un script shell simple.
- Exécuter efficacement les commandes sur des listes d'éléments dans un script ou à partir de la ligne de commande en utilisant des boucles et des conditions.
- Rechercher le texte correspondant à un motif dans les fichiers journaux et les sortie de commande à l'aide de la commande **grep** et des expressions régulières.

### SECTIONS

- Écriture de scripts bash simples (et exercice guidé)
- Exécution plus efficace des commandes à l'aide de boucles (et exercice guidé)
- Correspondance du texte dans les sorties de commande avec des expressions régulières (et exercice guidé)

### ATELIER

Amélioration de la productivité de la ligne de commande

# ÉCRITURE DE SCRIPTS BASH SIMPLES

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir automatiser des séquences de commandes en écrivant un script shell simple.

## CRÉATION ET EXÉCUTION DES SCRIPTS SHELL BASH

De nombreuses tâches d'administration système simples et courantes sont effectuées à l'aide d'outils de ligne de commande. Les tâches plus complexes nécessitent souvent le chaînage de plusieurs commandes qui se transmettent les résultats. À l'aide de l'environnement du shell bash et des fonctions de script, les commandes Linux sont combinées en *scripts shell* pour résoudre facilement des problèmes concrets ardu et répétitifs.

Dans sa forme la plus simple, un script shell bash est un fichier exécutable contenant une liste de commandes et éventuellement une logique de programmation permettant de contrôler la prise de décision dans une tâche globale. Lorsqu'il est bien écrit, un script shell est en lui-même un puissant outil de ligne de commande qui peut être exploité par d'autres scripts.

La maîtrise de la création de script shell est essentielle à une bonne administration système dans tout environnement opérationnel. Une connaissance pratique de la création de scripts shell est très importante dans les environnements entreprise où l'utilisation de scripts peut améliorer l'efficacité et la précision d'exécution de tâches routinières.

Vous pouvez créer un script shell bash en ouvrant un nouveau fichier vide dans un éditeur de texte. Bien que vous puissiez utiliser n'importe quel éditeur de texte, les éditeurs avancés tels que **vim** ou **emacs** comprennent la syntaxe shell bash et proposent une mise en surbrillance par **code de couleurs**. Cette mise en surbrillance permet d'identifier les erreurs courantes telles que la syntaxe incorrecte, les guillemets dépareillés, les parenthèses, les accolades et les crochets non fermés, etc.

### Spécification de l'interpréteur de commandes

La première ligne d'un script commence par la notation '#!', couramment appelée **sh-bang** ou **she-bang**, composée des noms de ces deux caractères, **sharp** (dièse) et **bang** (négation). Cette notation spécifique de **nombre magique** sur deux octets indique un script d'interprétation ; la syntaxe qui suit la notation est le nom complet de fichier qui permet à l'**interpréteur de commandes** approprié de traiter les lignes de ce script. Pour comprendre comment les **numéros magiques** indique les types de fichiers sous Linux, consultez les pages de manuel **file(1)** et **magic(5)**. Pour les fichiers script utilisant la syntaxe de script bash, la première ligne d'un script shell commence comme suit :

```
#!/bin/bash
```

### Exécution d'un script shell bash

Un script shell terminé doit pouvoir être exécuté comme une commande ordinaire. Utilisez la commande **chmod** pour ajouter l'autorisation d'exécution et associez-la éventuellement

## CHAPITRE 1 | Amélioration de la productivité de la ligne de commande

à la commande **chown** pour modifier le propriétaire du fichier script. Accordez l'autorisation d'exécution uniquement aux utilisateurs prévus du script.

Si vous placez le script dans l'un des répertoires listés dans la variable d'environnement PATH du shell, vous pouvez invoquer le script shell en utilisant le nom de fichier seul, comme pour toute autre commande. Le shell utilise la première commande trouvée avec ce nom de fichier ; évitez d'utiliser des noms de commande existants pour votre nom de fichier de script shell. Vous pouvez également invoquer un script shell en entrant un nom de chemin d'accès pour le script sur la ligne de commande. La commande **which**, suivie du nom de fichier du script exécutable, affiche le nom du chemin d'accès à la commande qui sera exécutée.

```
[user@host ~]$ which hello  
~/bin/Hello  
[user@host ~]$ echo $PATH  
/home/user/.local/bin:/home/user/bin:/usr/share/Modules/bin:/usr/local/bin:/usr/  
bin:/usr/local/sbin:/usr/sbin
```

## Encadrement des caractères spéciaux

Un certain nombre de caractères et de mots ont une signification particulière pour le shell bash. Cependant, il peut arriver que vous souhaitiez utiliser ces caractères pour leurs valeurs littérales, plutôt que pour leur signification particulière. Pour ce faire, utilisez l'un des trois outils pour supprimer (ou *neutraliser*) la signification spéciale : la barre oblique inverse (\), les guillemets simples ("") ou les guillemets doubles ("").

Le caractère d'échappement barre oblique inverse supprime la signification particulière du seul caractère qui le suit. Par exemple, pour afficher la chaîne de caractères littérale **# not a comment** à l'aide de la commande **echo**, le caractère **#** ne doit pas être interprété par bash pour sa signification particulière. Placez le caractère barre oblique inverse devant le caractère **#**.

```
[user@host ~]$ echo # not a comment  
  
[user@host ~]$ echo \# not a comment  
# not a comment
```

Lorsque plusieurs caractères d'une chaîne de texte doivent être ignorés, les utilisateurs peuvent utiliser plusieurs fois le caractère d'échappement ou utiliser des guillemets simples (""). Les guillemets simples conservent la signification littérale de tous les caractères qu'ils contiennent. Observez le caractère d'échappement et les guillemets simples en action :

```
[user@host ~]$ echo # not a comment #  
  
[user@host ~]$ echo \'# not a comment \'#  
# not a comment  
[user@host ~]$ echo \'# not a comment \'#  
# not a comment  
[user@host ~]$ echo '\"# not a comment \"'#  
# not a comment #
```

Utilisez les guillemets doubles pour supprimer la globalisation et l'extension par le shell, tout en permettant la substitution de commandes et de variables. La substitution de variables repose sur le même concept que la substitution de commandes, mais peut utiliser une syntaxe à base d'accolades. Observez les exemples de différentes formes d'utilisation des guillemets ci-dessous.

Utilisez des guillemets simples pour interpréter tout le texte littéralement. Outre la suppression de la globalisation et de l'extension par le shell, les guillemets indiquent au shell de supprimer la substitution de commandes et de variables. Le point d'interrogation (?) est un **métacaractère** qui doit également être protégé contre l'extension.

```
[user@host ~]$ var=$(hostname -s); echo $var
host
[user@host ~]$ echo "***** hostname is ${var} *****"
***** hostname is host *****
[user@host ~]$ echo Your username variable is \$USER.
Your username variable is $USER.
[user@host ~]$ echo "Will variable $var evaluate to $(hostname -s)?"
Will variable host evaluate to host?
[user@host ~]$ echo 'Will variable $var evaluate to $(hostname -s)?'
Will variable $var evaluate to $(hostname -s)?
[user@host ~]$ echo "\"Hello, world\""
"Hello, world"
[user@host ~]$ echo '"Hello, world"'
"Hello, world"
```

## Sortie à partir d'un script shell

La commande **echo** affiche un texte arbitraire en transmettant le texte en tant qu'argument à la commande. Par défaut, le texte s'affiche sur *sortie standard (STDOUT)*, mais il peut également être dirigé vers *erreur standard (STDERR)* à l'aide de la redirection du résultat. Dans le script bash simple suivant, la commande **echo** affiche le message "**Hello, world**" sur STDOUT.

```
[user@host ~]$ cat ~/bin/hello
#!/bin/bash

echo "Hello, world"

[user@host ~]$ hello
Hello, world
```



### NOTE

Cet utilisateur peut simplement exécuter **hello** à l'invite, car le répertoire **~/bin** (**/home/user/bin**) est dans la variable PATH de l'utilisateur et le script **hello** qui se trouve dedans est exécutable. Le shell trouve automatiquement le script à cet emplacement, tant qu'il n'existe aucun autre fichier exécutable appelé **hello** dans l'un des répertoires listés avant **/home/user/bin** dans la variable PATH.

La commande **echo** est largement utilisée dans les scripts shell pour afficher des messages d'information ou d'erreur. Ces messages peuvent être des indicateurs utiles sur la progression d'un script et peuvent également être dirigés vers sortie standard, erreur standard ou être redirigés vers un fichier journal pour y être archivés. Quand les messages d'erreur sont affichés, il est de bon usage de les diriger vers STDERR pour faciliter la différenciation des messages d'erreur des messages de statut normaux.

```
[user@host ~]$ cat ~/bin/hello
#!/bin/bash

echo "Hello, world"
echo "ERROR: Houston, we have a problem." >&2

[user@host ~]$ hello 2> hello.log
Hello, world
[user@host ~]$ cat hello.log
ERROR: Houston, we have a problem.
```

La commande **echo** peut aussi être très utile lors du débogage d'un script shell problématique. L'ajout d'instructions **echo** à la partie du script qui ne fonctionne pas comme prévu peut aider à clarifier les commandes exécutées ainsi que la valeur des variables invoquées.



## RÉFÉRENCES

Pages de manuel **bash(1)**, **magic(5)**, **echo(1)** et **echo(1p)**

## ► EXERCICE GUIDÉ

# ÉCRITURE DE SCRIPTS BASH SIMPLES

Dans cet exercice, vous allez écrire un script bash simple contenant une séquence de commandes et l'exécuter à partir de la ligne de commande.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Écrire et exécuter un script bash simple.
- Rediriger la sortie d'un script bash simple vers un fichier.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab console-write start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Le script vous alerte si elle n'est pas disponible. Il installe également le paquetage `vim-enhanced` si nécessaire.

```
[student@workstation ~]$ lab console-write start
```

- 1. À partir de `workstation`, ouvrez une session SSH sur `servera` en tant que `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Créez et exécutez un script bash simple.

- 2.1. Utilisez l'éditeur de texte `vim` pour créer un fichier texte sous votre répertoire personnel et nommez-le `firstscript.sh`

```
[student@servera ~]$ vim firstscript.sh
```

- 2.2. Insérez le texte suivant et enregistrez le fichier. Notez que le nombre de signes dièse (#) est arbitraire.

```
#!/bin/bash
echo "This is my first bash script" > ~/output.txt
echo "" >> ~/output.txt
echo "########################################" >> ~/output.txt
```

**CHAPITRE 1 |** Amélioration de la productivité de la ligne de commande

2.3. Utilisez la commande **sh** pour exécuter le script.

```
[student@servera ~]$ sh firstscript.sh
```

2.4. Consultez le fichier de sortie généré par le script.

```
[student@servera ~]$ cat output.txt
This is my first bash script

#####
```

- 3. Ajoutez d'autres commandes au script **firstscript.sh**, exécutez-le et passez en revue le résultat.

3.1. Utilisez l'éditeur de texte **vim** pour modifier le fichier **firstscript.sh**

```
[student@servera ~]$ vim firstscript.sh
```

3.2. Ajoutez les lignes suivantes en gras au fichier **firstscript.sh**.

```
#!/bin/bash
#
echo "This is my first bash script" > ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "LIST BLOCK DEVICES" >> ~/output.txt
echo "" >> ~/output.txt
lsblk >> ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "FILESYSTEM FREE SPACE STATUS" >> ~/output.txt
echo "" >> ~/output.txt
df -h >> ~/output.txt
echo "#####" >> ~/output.txt
```

3.3. Rendez le fichier **firstscript.sh** exécutable à l'aide de la commande **chmod**.

```
[student@servera ~]$ chmod a+x firstscript.sh
```

3.4. Exécutez le script **firstscript.sh**.

```
[student@servera ~]$ ./firstscript.sh
```

3.5. Examinez le fichier de sortie généré par le script.

```
[student@servera ~]$ cat output.txt
This is my first bash script

#####
LIST BLOCK DEVICES
```

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0      11:0    1 1024M  0 rom
vda     252:0    0   10G  0 disk
└─vda1  252:1    0   10G  0 part /
vdb     252:16   0    5G  0 disk

#####
FILESYSTEM FREE SPACE STATUS

Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        892M   0  892M  0% /dev
tmpfs          915M   0  915M  0% /dev/shm
tmpfs          915M  17M  899M  2% /run
tmpfs          915M   0  915M  0% /sys/fs/cgroup
/dev/vda1       10G  1.5G  8.6G 15% /
tmpfs         183M   0  183M  0% /run/user/1000
#####
```

- 4. Supprimez les fichiers d'exercice et déconnectez-vous de servera.

4.1. Supprimez le fichier script **firstscript.sh** et le fichier de sortie **output.txt**.

```
[student@servera ~]$ rm firstscript.sh output.txt
```

4.2. Déconnectez-vous de servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Terminer

Sur workstation, exécutez le script **lab console-write finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab console-write finish
```

L'exercice guidé est maintenant terminé.

# EXÉCUTION PLUS EFFICACE DES COMMANDES À L'AIDE DE BOUCLES

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Itérer sur des listes en utilisant les boucles **for**.
- Évaluer les codes de sortie des commandes et des scripts.
- Effectuer des tests en utilisant des opérateurs.
- Créer des structures conditionnelles en utilisant des instructions **if**.

## UTILISATION DE BOUCLES POUR ITÉRER DES COMMANDES

Les administrateurs système rencontrent souvent des tâches répétitives dans leurs activités quotidiennes. Les tâches répétitives peuvent prendre la forme d'une action à exécuter plusieurs fois sur une cible, comme la vérification d'un processus chaque minute pendant 10 minutes pour voir s'il est terminé. La répétition de tâches peut également prendre la forme d'une action à exécuter une fois sur plusieurs cibles, comme la création d'une sauvegarde de chaque base de données du système. La *boucle for* est l'une des constructions d'exécution en boucle du shell proposées par bash, qu'il est possible d'utiliser comme itérations de tâches.

### Traitement des éléments à partir de la ligne de commande

La construction en boucle for de bash utilise la syntaxe suivante.

```
for VARIABLE in LIST; do
    COMMAND VARIABLE
done
```

La boucle traite les chaînes fournies dans *LIST* dans l'ordre et une par une, puis se ferme une fois la dernière chaîne de la liste traitée. Chaque chaîne de la liste est enregistrée temporairement comme valeur de *VARIABLE*, tandis que la boucle **for** exécute le bloc de commandes contenu dans sa construction. Le nom de la variable est arbitraire. Généralement, la valeur de la variable est référencée par des commandes dans le bloc de commandes.

La liste des chaînes destinée à la boucle **for** peut être fournie de différentes manières. Il peut s'agir d'une liste de chaînes saisie directement par l'utilisateur ou créée à partir de différents types d'extension du shell tels qu'une variable, des accolades, l'extension du nom de fichier ou la substitution de commandes. Ci-dessous se trouvent quelques exemples illustrant les différentes manières d'ajouter des chaînes aux boucles **for**.

```
[user@host ~]$ for HOST in host1 host2 host3; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for HOST in host{1,2,3}; do echo $HOST; done
host1
```

```

host2
host3
[user@host ~]$ for HOST in host{1..3}; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for FILE in file*; do ls $FILE; done
filea
fileb
filec
[user@host ~]$ for FILE in file{a..c}; do ls $FILE; done
filea
fileb
filec
[user@host ~]$ for PACKAGE in $(rpm -qa | grep kernel); \
do echo "$PACKAGE was installed on \
$(date -d @$($PACKAGE --qf "%{INSTALLTIME}\n" $PACKAGE))"; done
abrt-addon-kerneloops-2.1.11-12.el7.x86_64 was installed on Tue Apr 22 00:09:07
EDT 2014
kernel-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:27:52 EDT 2014
kernel-tools-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:28:01 EDT 2014
kernel-tools-libs-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:26:22 EDT
2014
[user@host ~]$ for EVEN in $(seq 2 2 10); do echo "$EVEN"; done
2
4
6
8
10

```

## UTILISATION DES CODES DE SORTIE DANS UN SCRIPT

Une fois qu'un script a traité tout son contenu, il arrête le processus qui l'a appelé. Cependant, il peut être souhaitable de fermer un script avant la fin, lorsqu'une condition d'erreur se produit par exemple. Vous pouvez le faire en utilisant la commande **exit** dans un script. Quand un script rencontre la commande **exit**, il se ferme immédiatement et ne traite pas le reste du script.

La commande **exit** peut être exécutée à l'aide d'un argument optionnel sous forme de nombre entier compris entre **0** et **255** qui représente un code de sortie. Un code de sortie est un code qui est renvoyé après la fin d'un processus. Un code de sortie égal à **0** signifie qu'il n'y a aucune erreur. Toutes les autres valeurs indiquent un code de sortie d'erreur. Vous pouvez utiliser des valeurs différentes de zéro pour différencier les différents types d'erreurs rencontrés. Ce code de sortie est transmis au processus parent qui l'enregistre dans la variable **?** et devient accessible à l'aide du code **\$?**, comme le montrent les exemples suivants.

```

[user@host bin]$ cat hello
#!/bin/bash
echo "Hello, world"
exit 0

[user@host bin]$ ./hello
Hello, world

```

```
[user@host bin]$ echo $?  
0
```

Si la commande **exit** est appelée sans argument, le script se ferme et transmet le statut de sortie de la dernière commande exécutée au processus parent.

## TEST DES ENTRÉES DU SCRIPT

Pour s'assurer que les scripts ne soient pas facilement perturbés en cas de conditions inattendues, il convient de ne pas émettre d'hypothèse sur les entrées telles que les arguments de ligne de commande, les entrées utilisateur, les substitutions de commandes, les extensions de variables et les extensions de nom de fichier. L'intégrité peut être contrôlée à l'aide de la commande **test** de bash.

Comme toutes les commandes, la commande **test** produit un code de sortie une fois terminée, et ce code est enregistré en tant que valeur **\$?**. Pour voir la conclusion d'un test, affichez simplement la valeur de la variable **\$?** immédiatement après avoir exécuté la commande **test**. Un statut de sortie égal à **0** indique que le test a réussi, tandis qu'une valeur différente de zéro indique que le test a échoué.

Les tests sont effectués en utilisant une variété d'*opérateurs*. Les opérateurs peuvent être utilisés pour déterminer si un nombre est supérieur, supérieur ou égal à, inférieur à, inférieur ou égal à, ou égal à un autre nombre. Ils peuvent servir à vérifier si une chaîne de texte est identique ou non identique à une autre chaîne de texte. Les opérateurs peuvent également être utilisés pour évaluer si une variable a une valeur ou non.



### NOTE

De nombreux types d'*opérateurs* sont utilisés dans les scripts shell, en plus des opérateurs de comparaison enseignés ici. La page de manuel de **test(1)** liste les opérateurs importants d'*expression conditionnelle* accompagnés de leurs descriptions. La page de manuel de **bash(1)** explique également l'utilisation et l'évaluation de l'*opérateur*, mais elle est très difficile pour les débutants. Il est recommandé aux stagiaires d'approfondir leurs besoins en matière de scripts shell avancés au moyen de livres et de cours consacrés à la programmation shell.

Les exemples suivants présentent l'utilisation de la commande **test** à l'aide des opérateurs de comparaison numérique de bash.

```
[user@host ~]$ test 1 -gt 0 ; echo $?  
0  
[user@host ~]$ test 0 -gt 1 ; echo $?  
1
```

Les tests peuvent être réalisés à l'aide de la syntaxe de commande de test de bash, **[ <TESTEXPRESSION> ]**. Ils peuvent également être réalisés à l'aide de la dernière syntaxe de la commande de test étendue **[[ <TESTEXPRESSION> ]]**, qui est disponible depuis la version 2.02 de bash et qui fournit des fonctions telles que la correspondance de modèles glob et la correspondance de modèles regex.

Les exemples suivants présentent l'utilisation de la syntaxe de commande de test et des opérateurs de comparaison numérique de bash.

```
[user@host ~]$ [ 1 -eq 1 ]; echo $?
0
[user@host ~]$ [ 1 -ne 1 ]; echo $?
1
[user@host ~]$ [ 8 -gt 2 ]; echo $?
0
[user@host ~]$ [ 2 -ge 2 ]; echo $?
0
[user@host ~]$ [ 2 -lt 2 ]; echo $?
1
[user@host ~]$ [ 1 -lt 2 ]; echo $?
0
```

Les exemples suivants présentent l'utilisation des opérateurs de comparaison de chaînes de caractères de bash.

```
[user@host ~]$ [ abc = abc ]; echo $?
0
[user@host ~]$ [ abc == def ]; echo $?
1
[user@host ~]$ [ abc != def ]; echo $?
0
```

Les exemples suivants présentent l'utilisation des opérateurs unaires des chaînes de caractères de bash.

```
[user@host ~]$ STRING=''; [ -z "$STRING" ]; echo $?
0
[user@host ~]$ STRING='abc'; [ -n "$STRING" ]; echo $?
0
```



#### NOTE

Les espaces entre les crochets du test sont obligatoires car ils séparent les mots et les éléments de l'expression du test. La routine d'analyse des commandes du shell divise toutes les lignes de commande en mots et en opérateurs en reconnaissant les espaces et autres métacaractères, à l'aide de règles d'analyse intégrées. Pour un traitement complet de ce concept avancé, voir la page de manuel **getopt(3)**. Le caractère de crochet gauche ([]) est lui-même un alias intégré pour la commande **test**. Les mots de shell, qu'il s'agisse de commandes, de sous-commandes, d'options, d'arguments ou d'autres éléments de token, sont toujours délimités par des espaces.

## STRUCTURES CONDITIONNELLES

Les scripts shell simples représentent une série de commandes qui sont exécutées du début à la fin. Les structures conditionnelles permettent aux utilisateurs d'intégrer des prises de décision dans les scripts shell pour que certaines sections du script soient exécutées uniquement si certaines conditions sont satisfaites.

## Utilisation de la construction if/then

La structure conditionnelle la plus simple dans bash est la construction if/then dont la syntaxe est la suivante.

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
fi
```

Grâce à cette construction, si une condition donnée est satisfaite, une ou plusieurs actions sont entreprises. Si la condition donnée n'est pas satisfaite, aucune action n'est entreprise. Les tests numériques, de chaînes de caractères et de fichiers présentés plus haut sont souvent utilisés pour tester les conditions des instructions **if/then**. L'instruction **fi** à la fin ferme la construction **if/then**. La section de code suivante présente l'utilisation d'une construction **if/then** pour démarrer le service psacct s'il n'est pas activé.

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [ $? -ne 0 ]; then
> sudo systemctl start psacct
> fi
```

## Utilisation de la construction if/then/else

La construction **if/then** peut être étendue pour que différents ensembles d'actions soient mis en œuvre selon qu'une condition est satisfaite ou non. Cela est possible grâce à la construction **if/then/else**.

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
else
    <STATEMENT>
    ...
    <STATEMENT>
fi
```

La section de code suivante présente l'utilisation d'une instruction **if/then/else** pour démarrer le service psacct s'il n'est pas activé et l'arrêter s'il est activé.

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [ $? -ne 0 ]; then
> sudo systemctl start psacct
> else
> sudo systemctl stop psacct
> fi
```

## Utilisation de la construction if/then/elif/then/else

Enfin, la construction **if/then/else** peut être étendue pour tester plusieurs conditions, exécutant un ensemble d'actions différent quand une condition est satisfaite. Cette construction est présentée dans l'exemple suivant :

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
elif <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
else
    <STATEMENT>
    ...
    <STATEMENT>
fi
```

Dans cette structure conditionnelle, bash teste les conditions dans l'ordre présenté. Dès qu'une condition testée est vraie, bash exécute les actions associées à la condition et ignore le reste de la structure conditionnelle. Si aucune des conditions n'est vraie, bash exécute les actions énumérées dans la clause **else**.

La section de code suivante présente l'utilisation d'une instruction **if/then/elif/then/else** pour exécuter le client **mysql** si le service **mariadb** est activé, pour exécuter le client **psql** si le service **postgresql** est activé ou pour exécuter le client **sqlite3** si les services **mariadb** et **postgresql** sont tous les deux désactivés.

```
[user@host ~]$ systemctl is-active mariadb > /dev/null 2>&1
MARIADB_ACTIVE=$?
[user@host ~]$ sudo systemctl is-active postgresql > /dev/null 2>&1
POSTGRESQL_ACTIVE=$?
[user@host ~]$ if [ "$MARIADB_ACTIVE" -eq 0 ]; then
> mysql
> elif [ "$POSTGRESQL_ACTIVE" -eq 0 ]; then
> psql
> else
> sqlite3
> fi
```



### RÉFÉRENCES

Page de manuel (1)**bash**

## ► EXERCICE GUIDÉ

# EXÉCUTION PLUS EFFICACE DES COMMANDES À L'AIDE DE BOUCLES

Dans cet exercice, vous allez utiliser des boucles pour imprimer efficacement le nom d'hôte à partir de plusieurs serveurs.

## RÉSULTATS

Vous devez pouvoir créer une boucle **for** pour itérer sur une liste d'éléments de la ligne de commande et dans un script shell.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab console-commands start**. La commande exécute un script de démarrage qui détermine si les hôtes servera et serverb sont accessibles sur le réseau. Le script vous alerte s'ils ne sont pas disponibles.

```
[student@workstation ~]$ lab console-commands start
```

- 1. Utilisez les commandes **ssh** et **hostname** pour imprimer le nom d'hôte de servera et serverb vers une sortie standard.

```
[student@workstation ~]$ ssh student@servera hostname
servera.lab.example.com
[student@workstation ~]$ ssh student@serverb hostname
serverb.lab.example.com
```

- 2. Créez une boucle **for** pour effectuer la même tâche plus efficacement.

```
[student@workstation ~]$ for HOST in servera serverb
do
ssh student@${HOST} hostname
done
servera.lab.example.com
serverb.lab.example.com
```

- 3. Créez un script shell pour exécuter la même boucle **for**.

- 3.1. Créez le script shell dans le répertoire **/home/student/bin**.

```
[student@workstation ~]$ mkdir ~/bin
```

3.2. Vérifiez que le répertoire nouvellement créé est dans votre variable d'environnement PATH.

```
[student@workstation ~]$ echo $PATH  
/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/usr/local/  
sbin:/usr/sbin
```

3.3. Créez un script shell sur **/home/student/bin/printhostname.sh** pour exécuter la boucle **for**. Utilisez la commande **cat** pour vérifier le contenu de **printhostname.sh**.

```
[student@workstation ~]$ vim ~/bin/printhostname.sh  
[student@workstation ~]$ cat ~/bin/printhostname.sh  
#!/bin/bash  
#Execute for loop to print server hostname.  
for HOST in servera serverb  
do  
    ssh student@$HOST hostname  
done  
exit 0
```

3.4. Assurez-vous que le script que vous venez de créer est exécutable.

```
[student@workstation ~]$ chmod +x ~/bin/printhostname.sh
```

3.5. Exécutez le script à partir de votre répertoire personnel.

```
[student@workstation ~]$ printhostname.sh  
servera.lab.example.com  
serverb.lab.example.com
```

3.6. Vérifiez que le code de sortie de votre script est 0.

```
[student@workstation ~]$ echo $?  
0
```

## Terminer

Sur workstation, exécutez le script **lab console-commands finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab console-commands finish
```

L'exercice guidé est maintenant terminé.

# CORRESPONDANCE DU TEXTE DANS LES SORTIES DE COMMANDE AVEC DES EXPRESSIONS RÉGULIÈRES



## OBJECTIFS

Après avoir terminé cette section, les stagiaires doivent pouvoir réaliser les tâches suivantes :

- Créer des expressions régulières correspondant aux données recherchées.
- Appliquer des expressions régulières aux fichiers texte à l'aide de la commande **grep**.
- Effectuer des recherches dans des fichiers et des données à partir de commandes dotées de pipes en utilisant **grep**.

## ÉCRITURE D'EXPRESSIONS RÉGULIÈRES

Les expressions régulières fournissent un mécanisme de filtrage par motif facilitant la recherche de contenu spécifique. Les commandes, **vim**, **grep** et **less** peuvent toutes utiliser des expressions régulières. Les langages de programmation tels que Perl, Python et C peuvent tous utiliser des expressions régulières lors de l'application de critères de filtrage par motif.

Les expressions régulières constituent un langage à part entière. Elles ont donc une syntaxe et des règles qui leur sont propres. Cette section aborde la syntaxe régissant la création d'expressions régulières, et présente quelques exemples d'application.

### Expression régulière simple

L'expression régulière la plus simple est une correspondance exacte. Une correspondance exacte existe lorsque les caractères de l'expression régulière correspondent au type et à l'ordre des données recherchées.

Supposons qu'un utilisateur recherche dans le fichier de données ci-après toutes les occurrences du motif **cat** :

```
cat
dog
concatenate
dogma
category
educated
boondoggle
vindication
chilidog
```

La chaîne **cat** est une correspondance exacte des lettres **c**, **a** et **t** sans autres caractères entre chaque lettre. En utilisant **cat** comme expression régulière pour effectuer une recherche dans le fichier précédent, les correspondances suivantes sont renvoyées :

```
cat
concatenate
category
educated
vindication
```

## Correspondance au début et à la fin d'une ligne

La section précédente présentait l'application d'une expression régulière offrant une correspondance exacte dans un fichier. Notez que cette expression régulière identifie la chaîne de recherche indépendamment de son emplacement sur la ligne : début, fin ou milieu de mot ou de ligne. Utilisez un *ancrage de ligne* pour contrôler l'emplacement où l'expression régulière doit rechercher une correspondance.

Pour effectuer une recherche au début d'une ligne, utilisez le caractère caret (^). Pour effectuer une recherche en fin de ligne, utilisez le signe dollar (\$).

En utilisant le même fichier que ci-dessus, l'expression régulière **^cat** identifie deux mots. L'expression régulière **\$cat** ne trouve aucune correspondance.

```
cat
dog
concatenate
dogma
category
educated
boondoggle
vindication
chilidog
```

Pour localiser les lignes du fichier qui se terminent par **dog**, utilisez cette expression exacte suivie d'un ancrage de fin de ligne pour créer l'expression régulière **dog\$**. L'application de **dog\$** au fichier produit ces deux correspondances :

```
dog
chilidog
```

Pour localiser le seul mot sur une ligne, utilisez à la fois les ancrages de début et de fin de ligne. Par exemple, pour localiser le mot **cat** lorsqu'il s'agit du seul mot sur une ligne, utilisez **^cat\$**.

```
cat dog rabbit
cat
horse cat cow
cat pig
```

## Ajout de caractères génériques et d'opérateurs de répétition aux expressions régulières

Les expressions régulières utilisent un point (.) pour identifier n'importe quel caractère à l'exception du caractère de nouvelle ligne. L'expression régulière **c.t** recherche une chaîne contenant un **c**, suivi d'un caractère unique, suivi d'un **t**. Des exemples de correspondances comprennent **cat**, **concatenate**, **vindication**, **c5t** et **c\$t**.

## CHAPITRE 1 | Amélioration de la productivité de la ligne de commande

En utilisant un caractère générique illimité, vous ne pouvez pas prévoir le caractère qui correspondrait au caractère générique. Pour identifier des caractères spécifiques, remplacez le caractère générique illimité par des caractères acceptables. Le remplacement de l'expression régulière par **c[aou]t** identifie des motifs commençant par **c**, suivi soit d'un **a**, d'un **o** ou d'un **u**, puis d'un **t**.

Les *opérateurs de répétition* sont fréquemment utilisés avec les caractères génériques. Les opérateurs de répétition s'appliquent au caractère qui les précède dans l'expression régulière. L'un des plus couramment utilisés est l'astérisque ou le caractère étoile (\*). Lorsqu'il est utilisé dans une expression régulière, cet opérateur de répétition signifie une correspondance avec zéro caractères ou plus de l'expression précédente. Vous pouvez utiliser \* avec des expressions, pas seulement des caractères. Par exemple, **c[aou]\*t**. L'expression régulière **c.\*t** identifie **cat**, **coat**, **culvert** ainsi que **ct** (aucun caractère entre le **c** et le **t**). Toute donnée commençant par **c**, puis aucun ou plusieurs caractères, se terminant par **t**.

Un autre type d'opérateur de répétition indique le nombre exact de caractères précédents que doit contenir le motif. Par exemple, on trouve un opérateur de répétition explicite dans '**c.\{2\}t**'. Cette expression régulière identifie tout mot commençant par **c**, suivi de deux caractères exactement et se terminant par **t**. '**c.\{2\}t**' identifie deux mots dans l'exemple ci-dessous :

```
cat
coat convert
cart covert
cypher
```



### NOTE

Il est recommandé de mentionner les expressions régulières entre apostrophes, car elles contiennent fréquemment des métacaractères du shell (tels que \$, \* et {}). Cela garantit que les caractères sont interprétés par la commande et non par le shell.



### NOTE

Ce cours a introduit deux systèmes distincts d'analyse de texte par métacaractère : le *filtrage par motif* du shell (également connu sous le nom de globalisation de fichiers ou d'extension de nom de fichier) et les *expressions régulières*. Dans la mesure où ces deux systèmes utilisent des métacaractères similaires, tels que l'astérisque (\*), tout en présentant des différences en ce qui concerne les règles et l'interprétation des métacaractères, ils peuvent prêter à confusion jusqu'à ce que chacun soit suffisamment maîtrisé.

Le filtrage par motif est une technique d'analyse de ligne de commande conçue pour spécifier facilement de nombreux noms de fichiers. Elle est principalement prise en charge uniquement pour représenter des motifs de nom de fichier sur la ligne de commande. Les expressions régulières sont conçues pour représenter toute forme ou tout motif dans les chaînes de texte, quel qu'en soit le niveau de complexité. Les expressions régulières sont prises en charge en interne par de nombreuses commandes de traitement de texte, telles que **grep**, **sed**, **awk**, **python**, **perl** et de nombreuses applications, avec quelques variations minimales liées aux commandes dans les règles d'interprétation.

## Expressions régulières

OPTION	DESCRIPTION
.	Le point (.) correspond à un seul caractère.
?	L'élément précédent est facultatif et sera identifié une fois au plus.
*	L'élément précédent sera identifié zéro ou plusieurs fois.
+	L'élément précédent sera identifié une ou plusieurs fois.
{n}	L'élément précédent est identifié exactement n fois.
{n,}	L'élément précédent est identifié n ou plusieurs fois.
{,m}	L'élément précédent est identifié n ou plusieurs fois au plus.
{n,m}	L'élément précédent est identifié au moins n fois, mais pas plus de m fois.
[:alnum:]	Caractères alphanumériques : '[:alpha:]' et '[:digit:]' ; dans les paramètres régionaux 'C' et le codage de caractères ASCII, c'est équivalent à '[0-9A-Za-z]'.
[:alpha:]	Caractères alphabétiques : '[:lower:]' et '[:upper:]' ; dans les paramètres régionaux 'C' et le codage de caractères ASCII, c'est équivalent à '[A-Za-z]'.
[:blank:]	Caractères vides : espace et tabulation.
[:cntrl:]	Caractères de contrôle. En ASCII, ces caractères ont les codes octaux 000 à 037 et 177 (DEL). Dans d'autres jeux de caractères, ce sont les caractères équivalents, le cas échéant.
[:digit:]	Chiffres : 0 1 2 3 4 5 6 7 8 9.
[:graph:]	Caractères graphiques : '[:alnum:]' et '[:punct:]'.
[:lower:]	Lettres minuscules ; dans les paramètres régionaux 'C' et le codage de caractères ASCII, il s'agit de a b c d e f g h i j k l m n o p q r s t u v w x y z.
[:print:]	Caractères imprimables : '[:alnum:]', '[:punct:]' et l'espace.
[:punct:]	Caractères de ponctuation ; dans les paramètres régionaux 'C' et le codage de caractères ASCII, il s'agit de ! " # \$ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ' {   } ~. Dans d'autres jeux de caractères, ce sont les caractères équivalents, le cas échéant.
[:space:]	Caractères d'espace : dans les paramètres régionaux 'C', il s'agit des tabulations, des nouvelles lignes, des tabulations verticales, des sauts de formulaire, des retours à la ligne et des espaces.
[:upper:]	Lettres majuscules : dans les paramètres régionaux 'C' et le codage de caractères ASCII, il s'agit de A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.
[:xdigit:]	Chiffres hexadécimaux : 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.

OPTION	DESCRIPTION
\b	Identifie la chaîne vide à la périphérie d'un mot.
\B	Identifie la chaîne vide à condition qu'elle ne se trouve pas à la périphérie d'un mot.
\<	Identifie la chaîne vide au début d'un mot.
\>	Identifie la chaîne vide à la fin d'un mot.
\w	Identifie ce qui compose le mot. Synonyme de '[_[:alnum:]]'.
\W	Identifie ce qui compose toute autre chaîne qu'un mot. Synonyme de '[^_[:alnum:]]'.
\s	Identifie l'espace blanc. Synonyme de '[[space:]]'.
\S	Identifie ce qui n'est pas un espace blanc. Synonyme de '[^[:space:]]'.

## UTILISATION D'EXPRESSIONS RÉGULIÈRES AVEC GREP

La commande **grep** fait partie intégrante de la distribution. Elle utilise des expressions régulières pour isoler les données correspondantes.

### Isolement des données en utilisant la commande grep

La commande **grep** fournit une expression régulière et le nom d'un fichier auquel cette expression doit être comparée.

```
[user@host ~]$ grep '^computer' /usr/share/dict/words
computer
computerese
computerise
computerite
computerizable
computerization
computerize
computerized
computerizes
computerizing
computerlike
computernik
computers
```



#### NOTE

Il est recommandé de mentionner les expressions régulières entre apostrophes, car elles contiennent fréquemment des métacaractères du shell (tels que \$, \* et {}). Cela garantit que les caractères sont interprétés par la commande **grep** et non par le shell.

**CHAPITRE 1 |** Amélioration de la productivité de la ligne de commande

La commande **grep** peut s'utiliser en conjonction avec d'autres commandes au moyen d'un opérateur pipe (`|`). Par exemple :

```
[root@host ~]# ps aux | grep chrony
chrony      662  0.0  0.1  29440  2468 ?          S     10:56   0:00 /usr/sbin/chronyd
```

## Options grep

La commande **grep** dispose de nombreuses options permettant de préciser la façon d'utiliser l'expression régulière fournie avec des données.

**Tableau des options grep courantes**

OPTION	FONCTION
<b>-i</b>	Utiliser l'expression régulière fournie sans demander à tenir compte de la casse (ignorer les distinctions majuscules-minuscules).
<b>-v</b>	Afficher uniquement les lignes qui ne correspondent pas à l'expression régulière.
<b>-r</b>	Appliquer la recherche de données correspondant à l'expression régulière de manière récursive à un groupe de fichiers ou de répertoires.
<b>-A NUMBER</b>	Afficher le <i>NOMBRE</i> de lignes qui suivent la correspondance avec l'expression régulière.
<b>-B NUMBER</b>	Afficher le <i>NOMBRE</i> de lignes qui précèdent la correspondance avec l'expression régulière.
<b>-e</b>	L'utilisation de plusieurs options <b>-e</b> permet de spécifier plusieurs expressions régulières qui seront appliquées avec un « OU » logique.

**grep** possède de nombreuses autres options. Utilisez la page **man** pour les rechercher.

## Exemples grep

Les exemples suivants utilisent des fichiers de configuration et des fichiers journaux variés.

Les expressions régulières sont sensibles à la casse par défaut. Utilisez l'option **-i** avec **grep** pour exécuter une recherche non sensible à la casse. L'exemple suivant recherche le motif **serverroot**.

```
[user@host ~]$ cat /etc/httpd/conf/httpd.conf
...output omitted...
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
```

```
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
...output omitted...
```

```
[user@host ~]$ grep -i serverroot /etc/httpd/conf/httpd.conf
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# ServerRoot: The top of the directory tree under which the server's
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# same ServerRoot for multiple httpd daemons, you will need to change at
ServerRoot "/etc/httpd"
```

Dans les cas où vous ne savez pas ce que vous recherchez, l'option **-v** est très utile. L'option **-v** affiche uniquement les lignes qui ne correspondent pas à l'expression régulière. Dans l'exemple suivant, toutes les lignes, quelle que soit la casse, qui ne contiennent pas l'expression régulière **server** sont renvoyées.

```
[user@host ~]$ cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com classroom
172.25.254.254 content.example.com content
172.25.254.254 materials.example.com materials
172.25.250.254 workstation.lab.example.com workstation
### rht-vm-hosts file listing the entries to be appended to /etc/hosts

172.25.250.10 servera.lab.example.com servera
172.25.250.11 serverb.lab.example.com serverb
172.25.250.254 workstation.lab.example.com workstation
```

```
[user@host ~]$ grep -v -i server /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

172.25.254.254 classroom.example.com classroom
172.25.254.254 content.example.com content
172.25.254.254 materials.example.com materials
172.25.250.254 workstation.lab.example.com workstation
### rht-vm-hosts file listing the entries to be appended to /etc/hosts

172.25.250.254 workstation.lab.example.com workstation
```

Pour consulter un fichier sans être gêné par les lignes de commentaires, utilisez l'option **-v**. Dans l'exemple ci-après, l'expression régulière identifie toutes les lignes qui commencent par un # ou par un ; (ces caractères indiquant généralement qu'une ligne doit être interprétée comme un commentaire). Ces lignes sont alors omises de la sortie.

**CHAPITRE 1 |** Amélioration de la productivité de la ligne de commande

```
[user@host ~]$ cat /etc/ethertypes
#
# Ethernet frame types
#      This file describes some of the various Ethernet
#      protocol types that are used on Ethernet networks.
#
# This list could be found on:
#          http://www.iana.org/assignments/ethernet-numbers
#          http://www.iana.org/assignments/ieee-802-numbers
#
# <name>    <hexnumber> <alias1>...<alias35> #Comment
#
IPv4        0800     ip ip4      # Internet IP (IPv4)
X25        0805
ARP        0806     ether-arp   #
FR_ARP      0808           # Frame Relay ARP          [RFC1701]
...output omitted...
```

```
[user@host ~]$ grep -v '^#[;]' /etc/ethertypes
IPv4        0800     ip ip4      # Internet IP (IPv4)
X25        0805
ARP        0806     ether-arp   #
FR_ARP      0808           # Frame Relay ARP          [RFC1701]
```

La commande **grep** avec l'option **-e** vous permet de rechercher plusieurs expressions régulières à la fois. L'exemple suivant utilise une combinaison de **less** et **grep** pour localiser toutes les occurrences de **pam\_unix**, **user root** et **Accepted publickey** dans le fichier journal **/var/log/secure**.

```
[root@host ~]# cat /var/log/secure | grep -e 'pam_unix' \
-e 'user root' -e 'Accepted publickey' | less
Mar 19 08:04:46 jegui sshd[6141]: pam_unix(sshd:session): session opened for user
root by (uid=0)
Mar 19 08:04:50 jegui sshd[6144]: Disconnected from user root 172.25.250.254 port
41170
Mar 19 08:04:50 jegui sshd[6141]: pam_unix(sshd:session): session closed for user
root
Mar 19 08:04:53 jegui sshd[6168]: Accepted publickey for student from
172.25.250.254 port 41172 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCT
+wowZLNzNlBT0
```

Pour rechercher du texte dans un fichier ouvert à l'aide de **vim** ou **less**, utilisez la barre oblique (/) et tapez le motif à trouver. Appuyez sur **Entrée** pour lancer la recherche. Appuyez sur **N** pour trouver la prochaine correspondance.

```
[root@host ~]# vim /var/log/boot.log
...output omitted...
[[0;32m OK ^[[0m Reached target Initrd Default Target.^M
Starting dracut pre-pivot and cleanup hook...^M
[[0;32m OK ^[[0m Started dracut pre-pivot and cleanup hook.^M
Starting Cleaning Up and Shutting Down Daemons...^M
Starting Plymouth switch root service...^M
```

```
Starting Setup Virtual Console...^M
[^[[[0;32m OK ^[[0m] Stopped target Timers.^M
[^[[[0;32m OK ^[[0m] Stopped dracut pre-pivot and cleanup hook.^M
[^[[[0;32m OK ^[[0m] Stopped target Initrd Default Target.^M
/Daemons
```

```
[root@host ~]# less /var/log/messages
...output omitted...
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8584] Loaded device
plugin: NMTeamFactory (/usr/lib64/NetworkManager/1.14.0-14.el8/libnm-device-
plugin-team.so)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8599] device (lo):
carrier: link connected
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8600] manager:
(lo): new Generic device (/org/freedesktop/NetworkManager/Devices/1)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8623] manager:
(ens3): new Ethernet device (/org/freedesktop/NetworkManager/Devices/2)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8653] device
(ens3): state change: unmanaged -> unavailable (reason 'managed', sys-iface-
state: 'external')
/device
```



## RÉFÉRENCES

Pages de manuel **regex(7)** et **grep(1)**

## ► EXERCICE GUIDÉ

# CORRESPONDANCE DU TEXTE DANS LES SORTIES DE COMMANDE AVEC DES EXPRESSIONS RÉGULIÈRES

Dans cet atelier, vous allez rechercher du texte dans les journaux système et la sortie des commandes afin de trouver des informations plus efficacement.

## RÉSULTATS

Vous devez pouvoir rechercher efficacement du texte dans les fichiers journaux et les fichiers de configuration.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab console-regex start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle installe également le paquetage `postfix`.

```
[student@workstation ~]$ lab console-regex start
```

- ▶ 1. Utilisez la commande `ssh` pour vous connecter à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande `sudo -i` pour basculer vers l'utilisateur `root`. Le mot de passe de l'utilisateur `student` est `student`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Le paquetage `postfix` a été installé aujourd'hui par le script `start`. Utilisez la commande `grep` pour trouver le GID et l'UID des groupes et utilisateurs `postfix` et `postdrop`. Pour limiter les résultats de la commande `grep`, affichez tous les journaux avec une heure de début (`Start Time`) spécifique.

- 3.1. Utilisez la commande `date` pour déterminer l'heure actuelle.

```
[root@servera ~]# date
Fri Mar 22 08:23:56 CET 2019
```

- 3.2. Utilisez la commande **grep** avec la date, l'heure de début et les options GID pour trouver le GID et l'UID de l'utilisateur **postfix** et **postdrop**. Le script set-up de l'atelier a été exécuté quelques minutes avant l'heure actuelle. Prenez cela en considération lorsque vous effectuez une recherche dans le fichier journal **/var/log/secure**.

```
[root@servera ~]# grep '^Mar 22 08:2.*GID' /var/log/secure
Mar 22 08:20:04 servera groupadd[2514]: group added to /etc/group: name=postdrop,
GID=90
Mar 22 08:20:04 servera groupadd[2514]: new group: name=postdrop, GID=90
Mar 22 08:20:04 servera groupadd[2520]: group added to /etc/group: name=postfix,
GID=89
Mar 22 08:20:04 servera groupadd[2520]: new group: name=postfix, GID=89
Mar 22 08:20:04 servera useradd[2527]: new user: name=postfix, UID=89, GID=89,
home=/var/spool/postfix, shell=/sbin/nologin
```

- 4. Modifiez votre expression régulière pour qu'elle détecte les deux premiers messages dans le fichier **/var/log/maillog**. Notez que dans cette recherche, vous n'utilisez pas le caractère caret (^), car vous ne recherchez pas le premier caractère d'une ligne.

```
[root@servera ~]# grep 'postfix' /var/log/maillog | head -n 2
Mar 22 08:21:02 servera postfix/postfix-script[3879]: starting the Postfix mail
system
Mar 22 08:21:02 servera postfix/master[3881]: daemon started -- version 3.3.1,
configuration /etc/postfix
```

- 5. Vous devez trouver le nom du répertoire **queue** du serveur **postfix**. Recherchez dans le fichier de configuration **/etc/postfix/main.cf** toutes les informations sur les files d'attente. Utilisez l'option **-i** afin d'ignorer les distinctions de casse.

```
[root@servera ~]# grep -i 'queue' /etc/postfix/main.cf
# testing. When soft_bounce is enabled, mail will remain queued that
# The queue_directory specifies the location of the Postfix queue.
queue_directory = /var/spool/postfix
# QUEUE AND PROCESS OWNERSHIP
# The mail_owner parameter specifies the owner of the Postfix queue
# is the Sendmail-compatible mail queue listing command.
# setgid_group: The group for mail submission and queue management
```

- 6. Vérifiez que **postfix** écrit des messages à **/var/log/messages**. Utilisez la commande **less** suivie du caractère barre oblique (/) pour effectuer une recherche dans le fichier. Appuyez sur **n** pour passer à la prochaine entrée qui correspond à la recherche. Utilisez **q** pour quitter la commande **less**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 22 07:58:04 servera systemd[1]: Started Postfix Mail Transport Agent.
...output omitted...
Mar 22 08:12:26 servera systemd[1]: Stopping Postfix Mail Transport Agent...
Mar 22 08:12:26 servera systemd[1]: Stopped Postfix Mail Transport Agent.
...output omitted...
/Postfix
```

- 7. Utilisez la commande **ps aux** pour vérifier que le serveur **postfix** est en cours d'exécution. Limitez les résultats de **ps aux** en la combinant avec la commande **grep**.

```
[root@servera ~]# ps aux | grep postfix
root      3881  0.0  0.2 121664  5364 ?          Ss   08:21   0:00 /usr/
libexec/postfix/master -w
postfix    3882  0.0  0.4 147284  9088 ?          S     08:21   0:00 pickup -l -t unix
-u
postfix    3883  0.0  0.4 147336  9124 ?          S     08:21   0:00 qmgr -l -t unix -
u
```

- 8. Vérifiez que les files d'attente **qmgr**, **cleanup** et **pickup** sont correctement configurées. Utilisez la commande **grep** avec l'option **-e** afin de détecter plusieurs entrées dans le même fichier. Le fichier de configuration est **/etc/postfix/master.cf**

```
[root@servera ~]# grep -e qmgr -e pickup -e cleanup /etc/postfix/master.cf
pickup    unix n      -      n      60      1      pickup
cleanup   unix n      -      n      -      0      cleanup
qmgr      unix n      -      n      300      1      qmgr
#qmgr    unix n      -      n      300      1      oqmgr
```

- 9. Déconnectez-vous de servera.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Terminer

Sur workstation, exéutez le script **lab console-regex finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab console-regex finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# AMÉLIORATION DE LA PRODUCTIVITÉ DE LA LIGNE DE COMMANDE

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez créer un script bash capable de filtrer et d'obtenir des informations pertinentes de différents hôtes.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un script bash et rediriger sa sortie vers un fichier.
- Utiliser des boucles pour simplifier votre code.
- Filtrer le contenu pertinent à l'aide de **grep** et des expressions régulières.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant qu'utilisateur **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab console-review start**. Cette commande exécute un script de démarrage qui détermine si les machines **workstation**, **servera** et **serverb** sont accessibles sur le réseau. Le script vous alerte si elles ne sont pas disponibles. Il installe également les paquetages **vim-enhanced** et **util-linux** si nécessaire, configure **sudo** et prépare le contenu de **/var/log/secure** sur **servera** et **serverb**.

```
[student@workstation ~]$ lab console-review start
```

1. Créez le fichier script **/home/student/bin/bash-lab** sur **workstation**.
2. Modifiez le fichier script que vous venez de créer pour vous conformer aux informations demandées suivantes des hôtes **servera** et **serverb**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Aucun mot de passe n'est requis.

COMMANDÉ OU FICHIER	CONTENU DEMANDÉ
<b>hostname -f</b>	Obtenir tous les résultats.
<b>echo "#####"</b>	Obtenir tous les résultats.
<b>lscpu</b>	Obtenir uniquement les lignes qui commencent par la chaîne <b>CPU</b> .
<b>echo "#####"</b>	Obtenir tous les résultats.

COMMANDÉ OU FICHIER	CONTENU DEMANDÉ
<b>/etc/selinux/config</b>	Ignorer les lignes vides. Ignorer les lignes commençant par #.
<b>echo "#####"</b>	Obtenir tous les résultats.
<b>/var/log/secure</b>	Obtenir toutes les entrées « Failed password ».
<b>echo "#####"</b>	Obtenir tous les résultats.

Enregistrez les informations requises dans les nouveaux fichiers **/home/student/output-servera** et **/home/student/output-serverb**.



### NOTE

Vous pouvez utiliser **sudo** sans fournir de mot de passe sur les hôtes **servera** et **serverb**. N'oubliez pas d'utiliser une boucle pour simplifier votre script. Vous pouvez également concaténer plusieurs commandes **grep** en utilisant le caractère pipe (|).

- Exécutez le script **/home/student/bin/bash-lab** et examinez le contenu de la sortie sur **workstation**.

## Évaluation

A partir de **workstation**, exécutez la commande **lab console-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab console-review grade
```

## Fin

Sur **workstation**, exécutez le script **lab console-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab console-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# AMÉLIORATION DE LA PRODUCTIVITÉ DE LA LIGNE DE COMMANDE

### LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez créer un script bash capable de filtrer et d'obtenir des informations pertinentes de différents hôtes.

### RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un script bash et rediriger sa sortie vers un fichier.
- Utiliser des boucles pour simplifier votre code.
- Filtrer le contenu pertinent à l'aide de **grep** et des expressions régulières.

### AVANT DE COMMENCER

Connectez-vous à **workstation** en tant qu'utilisateur **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab console-review start**. Cette commande exécute un script de démarrage qui détermine si les machines **workstation**, **servera** et **serverb** sont accessibles sur le réseau. Le script vous alerte si elles ne sont pas disponibles. Il installe également les paquetages **vim-enhanced** et **util-linux** si nécessaire, configure **sudo** et prépare le contenu de **/var/log/secure** sur **servera** et **serverb**.

```
[student@workstation ~]$ lab console-review start
```

1. Créez le fichier script **/home/student/bin/bash-lab** sur **workstation**.

- 1.1. Sur **workstation**, créez le dossier **/home/student/bin/** si nécessaire.

```
[student@workstation ~]$ mkdir -p /home/student/bin
```

- 1.2. Utilisez **vim** pour créer et modifier le fichier script **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

- 1.3. Insérez le texte suivant et enregistrez le fichier.

```
#!/bin/bash
```

- 1.4. Rendez le fichier script exécutable.

**CHAPITRE 1 |** Amélioration de la productivité de la ligne de commande

```
[student@workstation ~]$ chmod a+x ~/bin/bash-lab
```

2. Modifiez le fichier script que vous venez de créer pour vous conformer aux informations demandées suivantes des hôtes **servera** et **serverb**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Aucun mot de passe n'est requis.

COMMANDÉ OU FICHIER	CONTENU DEMANDÉ
<b>hostname -f</b>	Obtenir tous les résultats.
<b>echo "#####"</b>	Obtenir tous les résultats.
<b>lscpu</b>	Obtenir uniquement les lignes qui commencent par la chaîne <b>CPU</b> .
<b>echo "#####"</b>	Obtenir tous les résultats.
<b>/etc/selinux/config</b>	Ignorer les lignes vides. Ignorer les lignes commençant par #.
<b>echo "#####"</b>	Obtenir tous les résultats.
<b>/var/log/secure</b>	Obtenir toutes les entrées « Failed password ».
<b>echo "#####"</b>	Obtenir tous les résultats.

Enregistrez les informations requises dans les nouveaux fichiers **/home/student/output-servera** et **/home/student/output-serverb**.

**NOTE**

Vous pouvez utiliser **sudo** sans fournir de mot de passe sur les hôtes **servera** et **serverb**. N'oubliez pas d'utiliser une boucle pour simplifier votre script. Vous pouvez également concaténer plusieurs commandes **grep** en utilisant le caractère pipe (|).

- 2.1. Utilisez **vim** pour ouvrir et modifier le fichier script **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

- 2.2. Ajoutez les lignes suivantes en gras au fichier script **/home/student/bin/bash-lab**.

**NOTE**

Voici un exemple de réalisation du script demandé. Dans le script bash, vous pouvez adopter différentes approches et obtenir le même résultat.

```
#!/bin/bash
#
USR='student'
OUT='/home/student/output'
#
for SRV in servera serverb
do
ssh ${USR}@${SRV} "hostname -f" > ${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "lscpu | grep '^CPU'" >> ${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "grep -v '^$' /etc/selinux/config|grep -v '^#' >> ${OUT}-${SRV}"
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "sudo grep 'Failed password' /var/log/secure" >> ${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
done
```

3. Exécutez le script **/home/student/bin/bash-lab** et examinez le contenu de la sortie sur workstation.

3.1. À partir de workstation, exécutez le script **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ bash-lab
```

3.2. Examinez le contenu de **/home/student/output-servera** et **/home/student/output-serverb**.

```
[student@workstation ~]$ cat /home/student/output-servera
servera.lab.example.com
#####
CPU op-mode(s):      32-bit, 64-bit
CPU(s):                2
CPU family:            21
CPU MHz:              2294.670
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:28 servera sshd[3939]: Failed password for invalid user operator1
from 172.25.250.9 port 58382 ssh2
Mar 21 22:30:31 servera sshd[3951]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 58384 ssh2
Mar 21 22:30:34 servera sshd[3953]: Failed password for invalid user manager1 from
172.25.250.9 port 58386 ssh2
#####
```

```
[student@workstation ~]$ cat /home/student/output-serverb
serverb.lab.example.com
#####
CPU op-mode(s):      32-bit, 64-bit
CPU(s):                2
```

```
CPU family:          6
CPU MHz:           2294.664
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:37 serverb sshd[3883]: Failed password for invalid user operator1
from 172.25.250.9 port 39008 ssh2
Mar 21 22:30:39 serverb sshd[3891]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 39010 ssh2
Mar 21 22:30:43 serverb sshd[3893]: Failed password for invalid user manager1 from
172.25.250.9 port 39012 ssh2
#####
```

## Évaluation

À partir de workstation, exéutez la commande **lab console-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab console-review grade
```

## Fin

Sur workstation, exéutez le script **lab console-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab console-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Comment créer et exécuter des scripts bash simples.
- Comment utiliser des boucles pour parcourir (itérer) une liste d'éléments de la ligne de commande et dans un script shell
- Comment rechercher du texte dans les fichiers journaux et les fichiers de configuration à l'aide d'expressions régulières et **grep**.



## CHAPITRE 2

# PLANIFICATION DE TÂCHES À VENIR

### PROJET

Planifier l'exécution automatique des tâches dans le futur.

### OBJECTIFS

- Configurer une commande qui s'exécute une fois dans le futur.
- Planifier des commandes à exécuter de manière répétitive à l'aide du fichier crontab d'un utilisateur.
- Planifier des commandes à exécuter de manière répétitive à l'aide des répertoires et du fichier crontab du système.
- Activer et désactiver les minuteurs systemd, et configurer un minuteur qui gère les fichiers temporaires.

### SECTIONS

- Planification d'une tâche utilisateur différée (et exercice guidé)
- Planification des tâches utilisateur récurrentes (et exercice guidé)
- Planification des tâches système récurrentes (et exercice guidé)
- Gestion des fichiers temporaires (et exercice guidé)

### ATELIER

Planification de tâches à venir

# PLANIFICATION D'UNE TÂCHE UTILISATEUR DIFFÉRÉE

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir configurer une commande qui s'exécute une fois dans le futur.

## DESCRIPTION DES TÂCHES UTILISATEUR DIFFÉRÉES

Vous pouvez être amené à exécuter une commande ou un ensemble de commandes à un moment donné dans le futur. Exemples : une personne veut planifier l'envoi d'un e-mail à son patron ; l'administrateur système qui travaille sur la configuration d'un pare-feu et qui définit une tâche « safety » pour réinitialiser les paramètres du pare-feu dix minutes plus tard, sauf s'il désactive la tâche en amont.

Ces commandes planifiées sont souvent appelées *tâches* ou *travaux*, et le terme *différé* indique que ces tâches ou travaux vont s'exécuter dans le futur.

L'une des solutions qui permet aux utilisateurs de Red Hat Enterprise Linux de planifier des tâches différées est *at*. Le paquetage *at* fournit le démon système (*atd*) avec un ensemble d'outils de ligne de commande pour interagir avec le démon (**at**, **atq**, etc). Dans une installation Red Hat Enterprise Linux par défaut, le démon *atd* est installé et activé automatiquement.

Les utilisateurs (y compris `root`) peuvent placer des tâches dans la file d'attente du démon *atd* à l'aide de la commande **at**. Le démon *atd* propose 26 files d'attente, de **a** à **z**, les tâches des files d'attente en fin d'alphabet ayant pour le système une priorité moindre (ou des valeurs de *politesse* plus élevées, comme le montrera un prochain chapitre).

## Planification de tâches utilisateur différées

Utilisez la commande **at TIMESPEC** pour planifier une nouvelle tâche. La commande **at** lit ensuite les commandes à exécuter à partir du canal `stdin`. Lorsque vous rédigez manuellement vos commandes, vous pouvez terminer la saisie en appuyant sur **Ctrl+D**. Pour les commandes plus complexes sujettes aux erreurs typographiques, il est souvent plus facile d'utiliser la redirection d'entrée à partir d'un fichier de script, par exemple, **at now +5min < myscript**, plutôt que de saisir toutes les commandes manuellement dans une fenêtre de terminal.

L'argument **TIMESPEC** avec la commande **at** accepte de nombreuses combinaisons puissantes, ce qui permet aux utilisateurs de décrire le moment exact où une tâche doit être exécutée. En général, on commence par indiquer une heure, par exemple **02:00, 15:59** ou même **teatime**, suivie en option d'une date ou d'un nombre de jours à venir. Le texte qui suit présente quelques exemples de combinaisons possibles.

- **now +5min**
- **teatime tomorrow** (l'heure du thé correspond à **16:00**)
- **noon +4 days**
- **5pm august 3 2021**

Pour une liste complète des spécifications de temps valables, reportez-vous à la définition **timespec** indiquée dans les références.

## EXAMEN ET GESTION DES TÂCHES UTILISATEUR DIFFÉRÉES

Pour obtenir un aperçu des tâches en attente de l'utilisateur actuel, utilisez les commandes **atq** ou **at -l**.

```
[user@host ~]$ atq
① 28 ② Mon Feb 2 05:13:00 2015 ③ a ④ user
29 Mon Feb 3 16:00:00 2014 h user
27 Tue Feb 4 12:00:00 2014 a user
```

Dans la sortie précédente, chaque ligne représente une tâche différente, planifiée pour être exécutée ultérieurement.

- ① Le numéro de travail unique pour cette tâche.
- ② La date et l'heure d'exécution de la tâche planifiée.
- ③ Indique que la tâche est planifiée avec la file d'attente par défaut a. Différentes tâches peuvent être planifiées avec diverses files d'attente.
- ④ Le propriétaire de la tâche (et l'utilisateur sous le nom duquel la tâche sera exécutée).



### IMPORTANT

Les utilisateurs sans privilège peuvent uniquement afficher et contrôler leurs propres tâches. L'utilisateur **root** peut afficher et gérer toutes les tâches.

Pour examiner les commandes qui seront lancées à l'exécution d'une tâche, utilisez la commande **at -c *JOBNUMBER***. Cette commande affiche l'*environnement* de la tâche en cours de configuration. Elle reflète l'environnement de l'utilisateur qui a créé la tâche, au moment où il l'a créée. Suivent enfin les commandes à exécuter.

## Suppression de tâches

La commande **atrm *JOBNUMBER*** permet de supprimer une tâche planifiée. Supprimez la tâche devenue inutile, par exemple quand la configuration à distance d'un pare-feu a réussi et n'a plus besoin d'être réinitialisée.



### RÉFÉRENCES

Pages du manuel **at(1)** et **atd(8)**

**/usr/share/doc/at/timespec**

## ► EXERCICE GUIDÉ

# PLANIFICATION D'UNE TÂCHE UTILISATEUR DIFFÉRÉE

Dans cet exercice, vous allez utiliser la commande **at** pour en planifier plusieurs à exécuter à des moments précis dans le futur.

## RÉSULTATS

Vous serez en mesure de réaliser les tâches suivantes :

- Planifier l'exécution d'une tâche dans le futur à une heure spécifiée.
- Examiner les commandes exécutées par une tâche planifiée.
- Supprimer les tâches planifiées.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exéutez **lab scheduling-at start** pour démarrer l'exercice. Ce script garantit que l'environnement a été correctement nettoyé et configuré.

```
[student@workstation ~]$ lab scheduling-at start
```

- 1. À partir de **workstation**, ouvrez une session SSH sur **servera** en tant que **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Planifiez l'exécution d'une tâche dans trois minutes à compter de cet instant à l'aide de la commande **at**. La tâche doit enregistrer la sortie de la commande **date** dans **/home/student/myjob.txt**.

- 2.1. Utilisez la commande **echo** pour transmettre la chaîne **date >> /home/student/myjob.txt** comme entrée à la commande **at** afin que la tâche s'exécute dans trois minutes à compter de maintenant.

```
[student@servera ~]$ echo "date >> /home/student/myjob.txt" | at now +3min
warning: commands will be executed using /bin/sh
job 1 at Thu Mar 21 12:30:00 2019
```

- 2.2. Utilisez la commande **atq** pour lister les tâches planifiées.

```
[student@servera ~]$ atq
1 Thu Mar 21 12:30:00 2019 a student
```

**CHAPITRE 2 |** Planification de tâches à venir

- 2.3. Utilisez la commande **watch atq** pour contrôler la surveillance de la file d'attente des tâches différées en temps réel. La tâche est supprimée de la file d'attente après son exécution.

```
[student@servera ~]$ watch atq
Every 2.0s: atq      servera.lab.example.com: Thu Mar 21 12:30:00 2019
1 Thu Mar 21 12:30:00 2019 a student
```

La commande **watch** précédente met à jour la sortie de **atq** toutes les deux secondes, par défaut. Une fois la tâche différée supprimée de la file d'attente, appuyez sur **Ctrl+c** pour quitter **watch** et revenez à l'invite du shell.

- 2.4. Utilisez la commande **cat** pour vérifier que le contenu de **/home/student/myjob.txt** correspond à la sortie de la commande **date**.

```
[student@servera ~]$ cat myjob.txt
Thu Mar 21 12:30:00 IST 2019
```

La sortie précédente correspond à la sortie de la commande **date**, ce qui confirme que la tâche planifiée a été correctement exécutée.

- 3. Utilisez la commande **at** pour planifier de manière interactive une tâche avec la file d'attente **g** qui s'exécute à **teatime** (16:00). La tâche doit exécuter une commande qui imprime le message **It's teatime** dans **/home/student/tea.txt**. Les nouveaux messages doivent être ajoutés au fichier **/home/student/tea.txt**.

```
[student@servera ~]$ at -q g teatime
warning: commands will be executed using /bin/sh
at> echo "It's teatime" >> /home/student/tea.txt
at> Ctrl+d
job 2 at Thu Mar 21 16:00:00 2019
```

- 4. Utilisez la commande **at** pour planifier de manière interactive une autre tâche avec la file d'attente **b** qui s'exécute à **16:05**. La tâche doit exécuter une commande qui imprime le message **The cookies are good** dans **/home/student/cookies.txt**. Les nouveaux messages doivent être ajoutés au fichier **/home/student/cookies.txt**.

```
[student@servera ~]$ at -q b 16:05
warning: commands will be executed using /bin/sh
at> echo "The cookies are good" >> /home/student/cookies.txt
at> Ctrl+d
job 3 at Thu Mar 21 16:05:00 2019
```

- 5. Examinez les commandes dans les tâches en attente.

- 5.1. Utilisez la commande **atq** pour afficher les numéros des tâches en attente.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
```

## CHAPITRE 2 | Planification de tâches à venir

Notez les numéros de travail dans la sortie précédente. Ces numéros de travail peuvent varier sur votre système.

5.2. Utilisez la commande **at** pour afficher les commandes dans la tâche en attente n°2.

```
[student@servera ~]$ at -c 2
...output omitted...
echo "It's teatime" >> /home/student/tea.txt
marcinDELIMITER28d54caa
```

Notez que la tâche planifiée précédente exécute une commande **echo** qui ajoute le message **It's teatime** dans **/home/student/tea.txt**.

5.3. Utilisez la commande **at** pour afficher les commandes dans la tâche en attente n°3.

```
[student@servera ~]$ at -c 3
...output omitted...
echo "The cookies are good" >> /home/student/cookies.txt
marcinDELIMITER1d2b47e9
```

Notez que la tâche planifiée précédente exécute une commande **echo** qui ajoute le message **The cookies are good** dans **/home/student/cookies.txt**.

- 6. Utilisez la commande **atq** pour afficher le numéro d'une tâche qui s'exécute à **teatime** (16:00) et supprimez-le en utilisant la commande **atrm**.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
[student@servera ~]$ atrm 2
```

- 7. Vérifiez que la tâche planifiée pour être exécutée à **teatime** (16:00) n'existe plus.

- 7.1. Utilisez la commande **atq** pour afficher la liste des tâches en attente et vérifier que la tâche planifiée pour être exécutée à **teatime** (16:00) n'existe plus.

```
[student@servera ~]$ atq
3 Thu Mar 21 16:05:00 2019 b student
```

7.2. Déconnectez-vous de servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exédez **lab scheduling-at finish** pour mettre fin à l'exercice. Ce script supprime les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab scheduling-at finish
```

L'exercice guidé est maintenant terminé.

# PLANIFICATION DES TÂCHES UTILISATEUR RÉCURRENTES

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir planifier des commandes à exécuter de manière répétitive à l'aide du fichier crontab d'un utilisateur.

## DESCRIPTION DES TÂCHES UTILISATEUR RÉCURRENTES

Les tâches dont l'exécution est répétée sont appelées *tâches récurrentes*. Les systèmes Red Hat Enterprise Linux sont fournis avec un démon **crond**, présent dans le paquetage **cronie**, activé et démarré par défaut pour gérer tout particulièrement les tâches récurrentes. Le démon **crond** lit plusieurs fichiers de configuration (un par utilisateur) que l'on peut modifier avec la commande **crontab**, ainsi que par un ensemble de fichiers système. Ces fichiers de configuration offrent aux utilisateurs et aux administrateurs un contrôle très précis sur l'échéance de l'exécution de leurs tâches récurrentes.

Si une commande planifiée génère une sortie ou une erreur non redirigée, le démon **crond** tente d'envoyer cette sortie ou erreur par e-mail à l'utilisateur propriétaire de la tâche (sauf indication contraire) à l'aide du serveur de messagerie configuré sur le système. En fonction de l'environnement, cela peut nécessiter une configuration supplémentaire. La sortie ou l'erreur de la commande planifiée peut être redirigée vers différents fichiers.

## PLANIFICATION DES TÂCHES UTILISATEUR RÉCURRENTES

Les utilisateurs standard peuvent gérer leurs tâches à l'aide de la commande **crontab**. Cette commande peut être appelée de quatre manières différentes :

### Exemples de crontab

COMMANDÉ	VOCATION
<b>crontab -l</b>	Liste les tâches de l'utilisateur actuel.
<b>crontab -r</b>	Supprime toutes les tâches de l'utilisateur actuel.
<b>crontab -e</b>	Modifie les tâches de l'utilisateur actuel.
<b>crontab filename</b>	Supprime toutes les tâches et les remplace par les tâches lues dans <i>filename</i> . Si aucun fichier n'est spécifié, <b>stdin</b> est utilisé.



### NOTE

Le superutilisateur peut utiliser l'option **-u** avec la commande **crontab** pour gérer les tâches d'un autre utilisateur. Vous ne devez pas utiliser la commande **crontab** pour gérer des tâches système. Recourez plutôt aux méthodes décrites à la section suivante.

## DESCRIPTION DU FORMAT DE LA TÂCHE UTILISATEUR

La commande **crontab -e** invoque Vim par défaut, à moins que la variable d'environnement EDITOR ait été définie sur autre chose. Entrez une tâche par ligne. Les autres entrées valides incluent : les lignes vides, généralement pour faciliter la lecture ; les commentaires, identifiés par des lignes commençant par le signe numérique (#) ; et les variables d'environnement utilisant le format **NOM=valeur**, qui affecte toutes les lignes situées en dessous de la ligne où elles sont déclarées. Les paramètres de variable courants incluent la variable SHELL, qui indique quel shell utiliser pour interpréter les lignes restantes du fichier crontab ; et la variable MAILTO déterminant qui doit recevoir toute sortie envoyée par e-mail.



### IMPORTANT

L'envoi d'e-mails peut nécessiter une configuration supplémentaire du serveur de messagerie local ou du relais SMTP sur un système.

Les champs dans le fichier **crontab** apparaissent dans l'ordre suivant :

- Minutes
- Heures
- Jour du mois
- Mois
- Jour de la semaine
- Commande



### IMPORTANT

Quand les champs **Day of month** et **Day of week** sont tous deux différents de \*, la commande s'exécute lorsque la valeur de l'un ou l'autre de ces deux champs est satisfaite. Par exemple, pour lancer une commande le 15 de chaque mois et tous les vendredis à 12:15, utilisez le format de tâche suivant :

```
15 12 15 * Fri command
```

Les mêmes règles syntaxiques s'appliquent aux cinq premiers champs :

- \* indique que la commande doit être exécutée de façon « indifférente »/systématique.
- Une valeur numérique spécifie un nombre de minutes ou d'heures, une date ou un jour de la semaine. Pour les jours de la semaine, 0 correspond à dimanche, 1 à lundi, 2 à mardi, etc. 7 équivaut également à dimanche.
- **x-y** pour une plage de valeurs, de **x** à **y** inclus.
- **x, y** définit une liste de valeurs. Les listes peuvent également inclure des plages de valeurs. Par exemple, l'entrée **5,10-13,17** dans la colonne **Minutes** indique qu'une tâche doit s'exécuter 5, 10, 11, 12, 13 et 17 minutes après l'heure juste.

## CHAPITRE 2 | Planification de tâches à venir

- `*/x` indique un intervalle de répétition de `x`. Par exemple, `*/7` dans la colonne **Minutes** exécute une tâche toutes les sept minutes.

On peut aussi utiliser les abréviations anglaises sur trois lettres pour les mois et les jours de la semaine, par exemple, « Jan », « Feb » et « Mon » « Tue ».

Le dernier champ contient la commande à exécuter à l'aide du shell par défaut. La variable d'environnement **SHELL** peut être utilisée pour changer le shell de la commande planifiée. Si la commande contient un signe de pourcentage sans signe d'échappement (%), ce signe de pourcentage est alors traité comme un saut de ligne, et tout ce qui suit est envoyé à la commande sur **stdin**.

## Exemple de tâches utilisateur récurrentes

Cette section décrit quelques exemples de tâches récurrentes.

- La tâche suivante exécute la commande **/usr/local/bin/yearly\_backup** à 9:00 précises, le 2 février de chaque année.

```
0 9 2 2 * /usr/local/bin/yearly_backup
```

- La tâche suivante envoie au propriétaire de la tâche un e-mail contenant le mot **Chime**, toutes les cinq minutes entre 9:00 et 17:00, chaque vendredi de juillet.

```
*/5 9-16 * Jul 5 echo "Chime"
```

La plage d'heures précédente **9-16** signifie que le minuteur de tâches commence à la neuvième heure (09:00) et continue jusqu'à la fin de la seizeième heure (16:59). La tâche commence à s'exécuter à **09:00** avec la dernière exécution à **16:55**, car cinq minutes après **16:55**, il sera **17:00**, ce qui se situe après la plage d'heures donnée.

- La tâche suivante exécute la commande **/usr/local/bin/daily\_report** tous les jours de la semaine, deux minutes avant minuit.

```
58 23 * * 1-5 /usr/local/bin/daily_report
```

- La tâche suivante exécute la commande **mutt** pour envoyer le message **Checking in** au destinataire **boss@example.com** chaque jour ouvrable (du lundi au vendredi), à 9:00.

```
0 9 * * 1-5 mutt -s "Checking in" boss@example.com % Hi there boss, just checking in.
```



### RÉFÉRENCES

Pages de manuel **crond(8)**, **crontab(1)** et **crontab(5)**

## ► EXERCICE GUIDÉ

# PLANIFICATION DES TÂCHES UTILISATEUR RÉCURRENTES

Dans cet exercice, vous allez planifier des commandes pour qu'elles s'exécutent de manière répétitive en tant qu'utilisateur sans privilège, à l'aide de la commande **crontab**.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Planifier des tâches récurrentes à exécuter en tant qu'utilisateur sans privilège.
- Examinez les commandes exécutées par une tâche récurrente planifiée.
- Supprimer les tâches récurrentes planifiées.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez **lab scheduling-cron start** pour démarrer l'exercice. Ce script garantit que l'environnement a été correctement nettoyé et configuré.

```
[student@workstation ~]$ lab scheduling-cron start
```

- 1. À partir de **workstation**, ouvrez une session SSH sur **servera** en tant que **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Planifiez une tâche récurrente en tant que **student** qui ajoute la date et l'heure actuelles à **/home/student/my\_first\_cron\_job.txt** toutes les deux minutes entre 8:00 et 21:00. La tâche ne doit être exécutée que du lundi au vendredi, pas le samedi ou le dimanche.



### IMPORTANT

Si vous travaillez sur cet atelier en dehors du jour et de l'heure mentionnés dans les instructions précédentes, vous devez ajuster l'heure système et/ou la date en conséquence afin que la tâche s'exécute pendant que vous travaillez.

- 2.1. Utilisez la commande **crontab -e** pour ouvrir crontab à l'aide de l'éditeur de texte par défaut.

```
[student@servera ~]$ crontab -e
```

**CHAPITRE 2 |** Planification de tâches à venir

2.2. Insérez la ligne suivante.

```
*2 08-20 * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

2.3. Dans l'éditeur de texte, appuyez sur **Échap** et tapez :**wq** pour enregistrer les modifications et quitter l'éditeur. Lorsque l'éditeur se ferme, vous devez voir le résultat suivant:

```
...output omitted...
crontab: installing new crontab
[student@servera ~]$
```

La sortie précédente confirme que la tâche a été correctement planifiée.

- 3. Utilisez la commande **crontab -l** pour lister les tâches récurrentes planifiées. Examinez la commande que vous avez planifiée pour être exécutée en tant que tâche récurrente à l'étape précédente.

```
[student@servera ~]$ crontab -l
*2 08-20 * * * /usr/bin/date >> /home/student/my_first_cron_job.txt
```

Notez que la tâche planifiée précédente exécute la commande **/usr/bin/date** et ajoute sa sortie dans **/home/student/my\_first\_cron\_job.txt**.

- 4. Utilisez la commande **while** afin que l'invite du shell soit en veille jusqu'à la création du fichier **/home/student/my\_first\_cron\_job.txt**, résultat de l'exécution réussie de la tâche récurrente que vous avez planifiée. Attendez que votre invite de shell réapparaisse.

```
[student@servera ~]$ while ! test -f my_first_cron_job.txt; do sleep 1s; done
```

La commande **while** précédente utilise **! test -f** pour continuer à exécuter une boucle de **sleep 1s** commandes jusqu'à ce que le fichier **my\_first\_cron\_job.txt** soit créé dans le répertoire **/home/student**.

- 5. Utilisez la commande **cat** pour vérifier que le contenu de **/home/student/my\_first\_cron\_job.txt** correspond à la sortie de la commande **date**.

```
[student@servera ~]$ cat my_first_cron_job.txt
Fri Mar 22 13:56:01 IST 2019
```

La sortie précédente peut différer sur votre système.

- 6. Supprimez toutes les tâches récurrentes planifiées à exécuter en tant que **student**.

6.1. Utilisez la commande **crontab -r** pour supprimer toutes les tâches récurrentes planifiées pour **student**.

```
[student@servera ~]$ crontab -r
```

6.2. Utilisez la commande **crontab -l** pour vérifier qu'il n'existe aucune tâche récurrente pour **student**.

```
[student@servera ~]$ crontab -l  
no crontab for student
```

6.3. Déconnectez-vous de servera.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez **lab scheduling-cron finish** pour mettre fin à l'exercice. Ce script supprime les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab scheduling-cron finish
```

L'exercice guidé est maintenant terminé.

# PLANIFICATION DES TÂCHES SYSTÈME RÉCURRENTES

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir planifier des commandes à exécuter de manière répétitive à l'aide des répertoires et du fichier crontab du système.

## DESCRIPTION DES TÂCHES SYSTÈME RÉCURRENTES

Les administrateurs système doivent souvent exécuter des tâches récurrentes. La meilleure pratique consiste à exécuter ces travaux à partir de comptes système plutôt que de comptes d'utilisateurs. En d'autres termes, ne planifiez pas l'exécution de ces tâches à l'aide de la commande **crontab**, mais utilisez plutôt les fichiers crontab dans l'ensemble du système. Les entrées de tâches dans les fichiers crontab à l'échelle du système sont similaires à celles des entrées crontab des utilisateurs, à la seule différence que les fichiers crontab à l'échelle du système ont un champ supplémentaire avant le champ de commande qui contient l'utilisateur sous l'autorité duquel la commande doit être exécutée.

Le fichier **/etc/crontab** propose un diagramme syntaxique pratique dans les commentaires inclus.

```
# For details see man 4 crontabs

# Example of job definition:
# ----- minute (0 - 59)
# | ----- hour (0 - 23)
# | | ----- day of month (1 - 31)
# | | | ----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | ----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue ...
# | | | | |
# * * * * * user-name command to be executed
```

Les tâches système récurrentes sont définies dans deux emplacements : le fichier **/etc/crontab** et les fichiers du répertoire **/etc/cron.d/**. Vous devez toujours créer vos fichiers crontab personnalisés dans le répertoire **/etc/cron.d** pour planifier les tâches système récurrentes. Placez le fichier crontab personnalisé dans **/etc/cron.d** pour lui éviter d'être écrasé si une mise à jour du package est appliquée au fournisseur de **/etc/crontab**, qui peut écraser le contenu existant dans **/etc/crontab**. Les packages qui requiert des tâches système récurrentes placent leurs fichiers crontab dans **/etc/cron.d/** contenant les entrées de tâche. Les administrateurs utilisent également cet emplacement pour regrouper les tâches connexes dans un seul fichier.

Le système crontab inclut également des référentiels pour les scripts devant être exécutés chaque heure, jour, semaine et mois. Ces référentiels sont des répertoires appelés **/etc/cron.hourly/**, **/etc/cron.daily/**, **/etc/cron.weekly/** et **/etc/cron.monthly/**. Notez à nouveau que ces répertoires contiennent des scripts shell exécutables, et non pas des fichiers crontab.

**IMPORTANT**

N'oubliez pas de rendre exécutable tout script que vous placez dans ces répertoires. Si un script n'est pas exécutable, il ne fonctionne pas. Pour rendre un script exécutable, utilisez la commande **chmod +x script\_name**.

Une commande nommée **run-parts** appelée à partir du fichier **/etc/cron.d/0hourly** exécute les scripts **/etc/cron.hourly/\***. La commande **run-parts** exécute également les tâches quotidiennes, hebdomadaires et mensuelles, mais à partir d'un fichier de configuration différent, nommé **/etc/anacrontab**.

**NOTE**

Auparavant, un service distinct appelé anacron était utilisé pour gérer le fichier **/etc/anacrontab**, mais dans Red Hat Enterprise Linux 7 et version ultérieure, ce fichier est analysé par le service crond.

**/etc/anacrontab** a pour objet de s'assurer que les tâches importantes sont exécutées systématiquement, et non ignorées accidentellement parce qu'un système est éteint ou en veille prolongée au moment où une tâche devrait s'exécuter. Par exemple, si une tâche système qui s'exécute quotidiennement n'a pas été exécutée la dernière fois parce que le système était en train de redémarrer, elle est exécutée dès que le système est opérationnel. Toutefois, le démarrage de la tâche peut prendre plusieurs minutes, selon la valeur du paramètre **Delay in minutes** spécifiée pour la tâche dans **/etc/anacrontab**.

Il existe différents fichiers dans **/var/spool/anacron/** pour chaque tâche quotidienne, hebdomadaire et mensuelle pour déterminer si une tâche particulière a été exécutée. Lorsque crond commence une tâche de **/etc/anacrontab**, il met à jour les horodatages de ces fichiers. Le même horodatage est utilisé pour déterminer la dernière exécution d'une tâche. La syntaxe du fichier **/etc/anacrontab** diffère de celle des fichiers de configuration **crontab** standard. Il contient exactement quatre champs par ligne, comme suit.

- **Périodicité en nombre de jours**

Intervalle en jours pour la tâche qui s'exécute de manière répétitive. Ce champ accepte un entier ou une macro comme valeur. Par exemple, la macro **@daily** est équivalente à l'entier **1**, ce qui signifie que la tâche est exécutée quotidiennement. De même, la macro **@weekly** est équivalente à l'entier **7**, ce qui signifie que la tâche est exécutée hebdomadairement.

- **Délai en minutes**

Délai pendant lequel le démon crond doit attendre avant de démarrer la tâche.

- **Identifiant de tâche**

Le nom unique de la tâche est identifié comme dans les messages du journal.

- **Commande**

La commande à exécuter.

Le fichier **/etc/anacrontab** contient également des déclarations de variable d'environnement utilisant la syntaxe **NOM=valeur**. La variable **START\_HOURS\_RANGE** présente un intérêt particulier, car elle spécifie l'intervalle de temps pour l'exécution des tâches. Les tâches ne sont

pas démarrées en dehors de cette plage. Si un jour donné, une tâche ne s'exécute pas dans cet intervalle de temps, elle doit attendre le jour suivant pour être exécutée.

## PRÉSENTATION DU MINUTEUR SYSTEMD

Avec l'introduction de `systemd` dans Red Hat Enterprise Linux 7, vous disposez désormais d'une nouvelle fonction de planification : les *unités minuteur systemd*. Une unité minuteur `systemd` active une autre unité d'un type différent (comme un service) dont le nom d'unité correspond au nom de l'unité minuteur. L'unité minuteur permet une activation basée sur le minuteur d'autres unités. Pour faciliter le débogage, `systemd` enregistre les événements du minuteur dans les journaux système.

### Échantillon de l'unité minuteur

Le paquetage `sysstat` fournit une unité minuteur `systemd` appelée `sysstat-collect.timer` pour collecter des statistiques système toutes les 10 minutes. La sortie suivante montre les lignes de configuration de `/usr/lib/systemd/system/sysstat-collect.timer`.

```
...output omitted...
[Unit]
Description=Run system activity accounting tool every 10 minutes

[Timer]
OnCalendar=*:00/10

[Install]
WantedBy=sysstat.service
```

Le paramètre **OnCalendar=\*:00/10** signifie que cette unité minuteur active l'unité correspondante (`sysstat-collect.service`) toutes les 10 minutes. Cependant, vous pouvez spécifier des intervalles de temps plus complexes. Par exemple, la valeur **2019-03-\* 12:35,37,39:16** pour le paramètre **OnCalendar** permet à l'unité minuteur d'activer l'unité de service correspondante à **12:35:16, 12:37:16 et 12:39:16** tous les jours pendant tout le mois de mars 2019. Vous pouvez également spécifier des minuteurs relatifs à l'aide de paramètres tels que **OnUnitActiveSec**. Par exemple, l'option **OnUnitActiveSec=15min** permet de déclencher l'unité minuteur correspondante 15 minutes après la dernière fois où son unité correspondante a été activée.



#### IMPORTANT

Ne modifiez aucun fichier de configuration d'unité sous le répertoire `/usr/lib/systemd/system`, car toute mise à jour du paquetage fournis du fichier de configuration peut remplacer les modifications apportées à la configuration dans ce fichier. Faites une copie du fichier de configuration de l'unité que vous voulez modifier sous le répertoire `/etc/systemd/system`, puis modifiez la copie afin que les modifications de configuration que vous apportez pour une unité ne soient pas remplacées par une mise à jour du paquetage du fournisseur. Si deux fichiers portent le même nom sous les répertoires `/usr/lib/systemd/system` et `/etc/systemd/system`, `systemd` analyse le fichier sous le répertoire `/etc/systemd/system`.

Après avoir modifié le fichier de configuration de l'unité minuteur, utilisez la commande **systemctl daemon-reload** pour vous assurer que `systemd` est conscient des changements. Cette commande recharge la configuration du gestionnaire `systemd`.

```
[root@host ~]# systemctl daemon-reload
```

Après avoir rechargé la configuration du gestionnaire `systemd`, utilisez la commande `systemctl` suivante pour activer l'unité minuteur.

```
[root@host ~]# systemctl enable --now <unitname>.timer
```



## RÉFÉRENCES

Pages de manuel **crontab(5)**, **anacron(8)**, **anacrontab(5)**, **systemd.time(7)**, **systemd.timer(5)** et **crond(8)**

## ► EXERCICE GUIDÉ

# PLANIFICATION DES TÂCHES SYSTÈME RÉCURRENTES

Dans cet exercice, vous allez planifier l'exécution de commandes selon différentes planifications en ajoutant des fichiers de configuration aux répertoires crontab du système.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Planifier une tâche système récurrente pour compter le nombre d'utilisateurs actifs.
- Mettre à jour l'unité du minuteur `systemd` qui rassemble les données d'activité du système.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant que `student` avec le mot de passe `student`.

Sur `workstation`, exécutez **lab scheduling-system start** pour démarrer l'exercice. Ce script garantit que l'environnement a été correctement nettoyé et configuré.

```
[student@workstation ~]$ lab scheduling-system start
```

- 1. À partir de `workstation`, ouvrez une session SSH sur `servera` en tant que `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Utilisez la commande **sudo -i** pour basculer vers le compte d'utilisateur `root`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Planifiez une tâche système récurrente qui génère un message de journal indiquant le nombre d'utilisateurs actuellement actifs dans le système. La tâche doit être exécutée quotidiennement. Vous pouvez utiliser la commande **w -h | wc -l** pour récupérer le nombre d'utilisateurs actuellement actifs dans le système. De plus, utilisez la commande **logger** pour générer le message du journal.

- 3.1. Créez un fichier script **/etc/cron.daily/usercount** avec le contenu ci-dessous. Vous pouvez utiliser la commande **vi /etc/cron.daily/usercount** pour créer le fichier script.

```
#!/bin/bash
USERCOUNT=$(w -h | wc -l)
logger "There are currently ${USERCOUNT} active users"
```

- 3.2. Utilisez la commande **chmod** pour activer l'autorisation (**x**) d'exécution sur **/etc/cron.daily/usercount**.

```
[root@servera ~]# chmod +x /etc/cron.daily/usercount
```

- 4. Le paquetage **sysstat** fournit les unités **systemd** appelées **sysstat-collect.timer** et **sysstat-collect.service**. L'unité minuteur déclenche l'unité de service toutes les 10 minutes pour collecter les données d'activité du système à l'aide du script shell appelé **/usr/lib64/sa/sa1**. Assurez-vous que le paquetage **sysstat** est installé et modifiez le fichier de configuration de l'unité minuteur pour collecter les données d'activité du système toutes les deux minutes.

- 4.1. Utilisez la commande **yum** pour installer le paquetage **sysstat**.

```
[root@servera ~]# yum install sysstat
...output omitted...
Is this ok [y/N]: y
...output omitted...
Installed:
  sysstat-11.7.3-2.el8.x86_64           lm_sensors-
  libs-3.4.0-17.20180522git70f7e08.el8.x86_64

Complete!
```

- 4.2. Copiez **/usr/lib/systemd/system/sysstat-collect.timer** dans **/etc/systemd/system/sysstat-collect.timer**.

```
[root@servera ~]# cp /usr/lib/systemd/system/sysstat-collect.timer \
/etc/systemd/system/sysstat-collect.timer
```



### IMPORTANT

Vous ne devez pas éditer de fichiers sous le répertoire **/usr/lib/systemd**. Avec **systemd**, vous pouvez copier le fichier d'unité dans le répertoire **/etc/systemd/system** et modifier cette copie. Le processus **systemd** analyse votre copie personnalisée à la place du fichier sous le répertoire **/usr/lib/systemd**.

- 4.3. Modifiez **/etc/systemd/system/sysstat-collect.timer** de sorte que le minuteur fonctionne toutes les deux minutes. En outre, remplacez toutes les occurrences de la chaîne **10 minutes** par **2 minutes** dans l'ensemble du fichier de configuration de l'unité, y compris celles des lignes commentées. Vous pouvez utiliser la commande **vi /etc/systemd/system/sysstat-collect.timer** pour éditer le fichier de configuration.

```
...
#      Activates activity collector every 2 minutes
```

```
[Unit]
Description=Run system activity accounting tool every 2 minutes

[Timer]
OnCalendar=*:00/02

[Install]
WantedBy=sysstat.service
```

Les changements précédents font en sorte que l'unité `sysstat-collect.timer` déclenche l'unité `sysstat-collect.service` toutes les deux minutes, qui exécute `/usr/lib64/sa/sa1 1 1`. L'exécution de `/usr/lib64/sa/sa1 1 1` collecte les données d'activité du système dans un fichier binaire sous le répertoire `/var/log/sa`.

- 4.4. Utilisez la commande `systemctl daemon-reload` pour vous assurer que `systemd` est conscient des changements.

```
[root@servera ~]# systemctl daemon-reload
```

- 4.5. Utilisez la commande `systemctl` pour activer l'unité minuteur `sysstat-collect.timer`.

```
[root@servera ~]# systemctl enable --now sysstat-collect.timer
```

- 4.6. Utilisez la commande `while` pour patienter le temps que le fichier binaire soit créé sous le répertoire `/var/log/sa`. Attendez que votre invite de shell réapparaisse.

```
[root@servera ~]# while [ $(ls /var/log/sa | wc -l) -eq 0 ]; do sleep 1s; done
```

Dans la commande `while` ci-dessus, la commande `ls /var/log/sa | wc -l` renvoie un **0** si le fichier n'existe pas et un **1** s'il existe. La commande `while` détermine si cela est égal à **0** et si oui, entre dans la boucle, qui marque une pause d'une seconde. Lorsque le fichier existe, la boucle `while` se ferme.

- 4.7. Utilisez la commande `ls -l` pour vérifier que le fichier binaire sous le répertoire `/var/log/sa` a été modifié lors des deux dernières minutes.

```
[root@servera ~]# ls -l /var/log/sa
total 8
-rw-r--r--. 1 root root 5156 Mar 25 12:34 sa25
[root@servera ~]# date
Mon Mar 25 12:35:32 +07 2019
```

La sortie des commandes précédentes peut différer sur votre système.

- 4.8. Quittez le shell `root` de l'utilisateur.

```
[root@servera ~]# exit
logout
[student@servera ~]$
```

- 4.9. Déconnectez-vous de `servera`.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez **lab scheduling-system finish** pour mettre fin à l'exercice. Ce script supprime les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab scheduling-system finish
```

L'exercice guidé est maintenant terminé.

# GESTION DES FICHIERS TEMPORAIRES

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir activer et désactiver les minuteurs `systemd`, et configurer un minuteur qui gère les fichiers temporaires.

## GESTION DES FICHIERS TEMPORAIRES

Un système moderne requiert un grand nombre de fichiers et de répertoires temporaires. Quelques applications (et utilisateurs) conservent leurs données temporaires dans le répertoire `/tmp`, tandis que d'autres utilisent un emplacement propre aux tâches tel que les répertoires `volatile` du démon et les répertoires propres à l'utilisateur sous `/run`. Dans ce contexte, le terme « `volatile` » signifie que le système de fichiers qui contient ces fichiers existe uniquement en mémoire. Lorsque le système redémarre ou n'est plus alimenté, tout le contenu de la mémoire `volatile` est perdu.

Pour garder un système en bon état de fonctionnement, il faut créer ces répertoires et fichiers s'ils n'existent pas, car certains démons et scripts pourraient dépendre de leur présence et purger les anciens fichiers pour éviter de saturer les disques ou de fournir des informations erronées.

Red Hat Enterprise Linux 7 et version ultérieure incluent un nouvel outil appelé **`systemd-tmpfiles`**, qui fournit une méthode structurée et configurable pour gérer les répertoires et les fichiers temporaires.

Lorsque `systemd` démarre un système, l'une des premières unités de service lancées est `systemd-tmpfiles-setup`. Ce service exécute la commande `systemd-tmpfiles --create --remove`. Cette commande lit les fichiers de configuration à partir de `/usr/lib/tmpfiles.d/* .conf`, `/run/tmpfiles.d/* .conf` et `/etc/tmpfiles.d/* .conf`. Tout fichier ou répertoire dont la suppression est inscrite dans ces fichiers de configuration est supprimé, et tout fichier ou répertoire dont la création (ou la correction des permissions) est inscrite sera créé avec les permissions adéquates, si nécessaire.

## Nettoyage de fichiers temporaires avec un minuteur `systemd`

Pour garantir que les systèmes à exécution longue ne remplissent pas leurs disques avec des données obsolètes, une unité minuteur `systemd` appelée `systemd-tmpfiles-clean.timer` déclenche `systemd-tmpfiles-clean.service` à intervalle régulier, qui exécute la commande `systemd-tmpfiles --clean`.

Les fichiers de configuration du minuteur `systemd` ont une section **[Timer]** qui indique la fréquence de démarrage du service du même nom.

Utilisez la commande `systemctl` suivante pour afficher le contenu du fichier de configuration de l'unité `systemd-tmpfiles-clean.timer`.

```
[user@host ~]$ systemctl cat systemd-tmpfiles-clean.timer
# /usr/lib/systemd/system/systemd-tmpfiles-clean.timer
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
```

```
#  
# systemd is free software; you can redistribute it and/or modify it  
# under the terms of the GNU Lesser General Public License as published  
# by  
# the Free Software Foundation; either version 2.1 of the License, or  
# (at your option) any later version.  
  
[Unit]  
Description=Daily Cleanup of Temporary Directories  
Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)  
  
[Timer]  
OnBootSec=15min  
OnUnitActiveSec=1d
```

Dans la configuration précédente, le paramètre **OnBootSec=15min** indique que l'unité de service appelée `systemd-tmpfiles-clean.service` se déclenche 15 minutes après le démarrage du système. Le paramètre **OnUnitActiveSec=1d** indique que tout autre déclenchement de l'unité de service `systemd-tmpfiles-clean.service` se produit 24 heures après sa dernière activation.

En fonction de vos besoins, vous pouvez modifier les paramètres du fichier de configuration du minuteur **systemd-tmpfiles-clean.timer**. Par exemple, la valeur **30 minutes** du paramètre **OnUnitActiveSec** déclenche l'unité de service `systemd-tmpfiles-clean.service` 30 minutes après sa dernière activation. Par conséquent, `systemd-tmpfiles-clean.service` se déclenche toutes les 30 minutes après l'application des modifications.

Après avoir modifié le fichier de configuration de l'unité minuteur, utilisez la commande **systemctl daemon-reload** pour vous assurer que `systemd` est conscient des changements. Cette commande recharge la configuration du gestionnaire `systemd`.

```
[root@host ~]# systemctl daemon-reload
```

Après avoir rechargé la configuration du gestionnaire `systemd`, utilisez la commande **systemctl** suivante pour activer l'unité `systemd-tmpfiles-clean.timer`.

```
[root@host ~]# systemctl enable --now systemd-tmpfiles-clean.timer
```

## Nettoyage manuel des fichiers temporaires

La commande **systemd-tmpfiles --clean** permet d'analyser les mêmes fichiers de configuration que la commande **systemd-tmpfiles --create**, mais au lieu de créer des fichiers et des répertoires, elle purge tous les fichiers qui n'ont pas été consultés, remplacés ou modifiés plus récemment que l'ancienneté maximale définie dans le fichier de configuration.

Le format des fichiers de configuration de la commande **systemd-tmpfiles** est détaillé dans la page de manuel **tmpfiles.d(5)**. La syntaxe de base comporte sept colonnes : Type, Path, Mode, UID, GID, Age et Argument. Type fait référence à l'action que la commande **systemd-tmpfiles** doit effectuer ; par exemple **d** pour créer un répertoire s'il n'existe pas déjà, ou **Z** pour restaurer de manière récursive les contextes SELinux, permissions et droits de propriété des fichiers.

Les exemples suivants comportent des explications.

```
d /run/systemd/seats 0755 root root -
```

Lors de la création de fichiers et de répertoires, créez, s'il n'existe pas déjà, le répertoire **/run/systemd/seats**, affecté à l'utilisateur **root** et au groupe **root**, avec les permissions définies sur **rwxr-xr-x**. Ce répertoire ne sera pas purgé automatiquement.

```
D /home/student 0700 student student 1d
```

Créez le répertoire **/home/student** s'il n'existe pas déjà. S'il existe, ce code vide l'intégralité de son contenu. Lorsque la commande **systemd-tmpfiles --clean** est exécutée, elle supprime tous les fichiers qui n'ont pas été consultés, remplacés ni modifiés depuis au moins une journée.

```
L /run/fstablink - root root - /etc/fstab
```

Créez le lien symbolique **/run/fstablink** pointant vers **/etc/fstab**. Ne purgez jamais cette ligne automatiquement.

## Ordre de priorité des fichiers de configuration

Les fichiers de configuration peuvent être stockés à trois emplacements :

- **/etc/tmpfiles.d/\* .conf**
- **/run/tmpfiles.d/\* .conf**
- **/usr/lib/tmpfiles.d/\* .conf**

Les fichiers du répertoire **/usr/lib/tmpfiles.d/** sont fournis par les paquetages RPM appropriés, et ne sont pas modifiables. Les fichiers sous **/run/tmpfiles.d/** sont eux-mêmes des fichiers volatiles, normalement utilisés par les démons pour gérer leurs propres fichiers temporaires d'exécution. Les fichiers sous **/etc/tmpfiles.d/** sont destinés aux administrateurs pour configurer des emplacements temporaires personnalisés et pour remplacer les valeurs par défaut définies par les fournisseurs.

Si un fichier dans **/run/tmpfiles.d/** porte le même nom qu'un fichier dans **/usr/lib/tmpfiles.d/**, c'est le fichier dans **/run/tmpfiles.d/** qui est utilisé. Si un fichier dans **/etc/tmpfiles.d/** porte le même nom qu'un fichier dans **/run/tmpfiles.d/** ou **/usr/lib/tmpfiles.d/**, c'est le fichier dans **/etc/tmpfiles.d/** qui est utilisé.

Compte tenu de ces règles de priorité, vous pouvez aisément outrepasser les réglages du fournisseur, en copiant le fichier approprié dans **/etc/tmpfiles.d/**, puis en le modifiant. Grâce à cette procédure, les paramètres définis par l'administrateur sont facilement gérables à partir d'un système centralisé de gestion de la configuration, et ne risquent pas d'être écrasés par la mise à jour d'un paquetage.



### NOTE

Quand on teste des configurations, nouvelles ou modifiées, il peut s'avérer utile de n'appliquer les commandes que depuis un seul fichier de configuration. Pour ce faire, il suffit de spécifier le nom du fichier de configuration sur la ligne de commande.



## RÉFÉRENCES

Pages man **systemd-tmpfiles(5)**, **tmpfiles.d(8)**, **stat(2)**, **stat(5)** et  
**systemd.timer(1)**

## ► EXERCICE GUIDÉ

# GESTION DES FICHIERS TEMPORAIRES

Dans cet exercice, vous allez configurer **systemd-tmpfiles** afin de changer la rapidité avec laquelle il supprime les fichiers temporaires de **/tmp**, et également pour purger périodiquement les fichiers d'un autre répertoire.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Configurer **systemd-tmpfiles** pour supprimer les fichiers temporaires inutilisés de **/tmp**.
- Configurer **systemd-tmpfiles** pour purger périodiquement les fichiers d'un autre répertoire.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez **lab scheduling-tempfiles start** pour démarrer l'exercice. Ce script crée les fichiers nécessaires et garantit que l'environnement est correctement configuré.

```
[student@workstation ~]$ lab scheduling-tempfiles start
```

- 1. À partir de **workstation**, ouvrez une session SSH sur **servera** en tant que **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Configurez **systemd-tmpfiles** pour qu'il nettoie le répertoire **/tmp** de sorte qu'il ne contienne aucun fichier qui n'aurait pas été utilisé au cours des cinq derniers jours. Assurez-vous que la configuration ne soit pas écrasée par une mise à jour de paquetage.

2.1. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2.2. Copiez **/usr/lib/tmpfiles.d/tmp.conf** dans **/etc/tmpfiles.d/tmp.conf**.

```
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/tmp.conf
```

- 2.3. Recherchez la ligne de configuration dans **/etc/tmpfiles.d/tmp.conf** qui s'applique au répertoire **/tmp**. Remplacez l'ancienneté existante des fichiers temporaires dans cette ligne de configuration par **5** jours. Supprimez toutes les autres lignes du fichier, y compris celles qui sont commentées. Vous pouvez utiliser la commande **vim /etc/tmpfiles.d/tmp.conf** pour éditer le fichier de configuration. Le fichier **/etc/tmpfiles.d/tmp.conf** doit se présenter comme suit :

```
q /tmp 1777 root root 5d
```

La configuration précédente fait en sorte que **systemd-tmpfiles** garantisse la définition du répertoire **/tmp** avec les permissions octales sur **1777**. L'utilisateur propriétaire et le groupe de **/tmp** doit être **root**. Le répertoire **/tmp** doit être exempt de fichiers temporaires inutilisés au cours des cinq derniers jours.

- 2.4. Utilisez la commande **systemd-tmpfiles --clean** pour vérifier que le fichier **/etc/tmpfiles.d/tmp.conf** contient la configuration appropriée.

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf
```

Comme la commande précédente n'a renvoyé aucune erreur, cela confirme que les paramètres de configuration sont corrects.

- 3. Ajoutez une nouvelle configuration garantissant que le répertoire **/run/momentary** est défini avec la propriété d'utilisateur et de groupe sur **root**. Les permissions octales du répertoire doivent être **0700**. La configuration doit purger tout fichier de ce répertoire qui reste inutilisé au cours des 30 dernières secondes.
- 3.1. Créez le fichier nommé **/etc/tmpfiles.d/momentary.conf** avec le contenu ci-dessous. Vous pouvez utiliser la commande **vim /etc/tmpfiles.d/momentary.conf** pour créer le fichier de configuration.

```
d /run/momentary 0700 root root 30s
```

La configuration précédente fait en sorte que **systemd-tmpfiles** garantisse la définition du répertoire **/run/momentary** avec ses permissions octales sur **0700**. La propriété d'utilisateur et groupe de **/run/momentary** doit être **root**. Tout fichier de ce répertoire qui reste inutilisé au cours des 30 dernières secondes doit être purgé.

- 3.2. Utilisez la commande **systemd-tmpfiles --create** pour vérifier que le fichier **/etc/tmpfiles.d/momentary.conf** contient la configuration appropriée. La commande crée le répertoire **/run/momentary** s'il n'existe pas déjà.

```
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/momentary.conf
```

Comme la commande précédente n'a renvoyé aucune erreur, cela confirme que les paramètres de configuration sont corrects.

- 3.3. Utilisez la commande **ls** pour vérifier que le répertoire **/run/momentary** est créé avec les permissions, le propriétaire et le propriétaire du groupe appropriés.

```
[root@servera ~]# ls -ld /run/momentary
drwx----- 2 root root 40 Mar 21 16:39 /run/momentary
```

Notez que le jeu de permissions octal de **/run/momentary** est **0700** et que la propriété d'utilisateur et de groupe est définie sur **root**.

- 4. Vérifiez que tout fichier sous le répertoire **/run/momentary** qui est resté inutilisé au cours des 30 dernières secondes est supprimé, en fonction de la configuration **systemd-tmpfiles** du répertoire.

- 4.1. Utilisez la commande **touch** pour créer un fichier appelé **/run/momentary/testfile**.

```
[root@servera ~]# touch /run/momentary/testfile
```

- 4.2. Utilisez la commande **sleep** pour configurer votre invite de shell de sorte qu'elle ne réapparaisse pas avant 30 secondes.

```
[root@servera ~]# sleep 30
```

- 4.3. Après le retour de votre invite de shell, utilisez la commande **systemd-tmpfiles --clean** pour nettoyer les fichiers obsolètes de **/run/momentary**, en fonction de la règle mentionnée dans **/etc/tmpfiles.d/momentary.conf**.

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/momentary.conf
```

La commande précédente supprime **/run/momentary/testfile**, car le fichier est resté inutilisé pendant 30 secondes et aurait dû être supprimé conformément à la règle mentionnée dans **/etc/tmpfiles.d/momentary.conf**.

- 4.4. Utilisez la commande **ls -l** pour vérifier que le fichier **/run/momentary/testfile** existe.

```
[root@servera ~]# ls -l /run/momentary/testfile
ls: cannot access '/run/momentary/testfile': No such file or directory
```

- 4.5. Quittez le shell de l'utilisateur **root** pour revenir à l'utilisateur **student**.

```
[root@servera ~]# exit
logout
[student@servera ~]$
```

- 4.6. Déconnectez-vous de **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur `workstation`, exécutez `lab scheduling-tempfiles finish` pour mettre fin à l'exercice. Ce script supprime les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab scheduling-tempfiles finish
```

L'exercice guidé est maintenant terminé.

## ► QUIZ

# PLANIFICATION DE TÂCHES À VENIR

Répondez aux questions suivantes en sélectionnant un ou plusieurs éléments.

- ▶ 1. Quelle commande affiche toutes les tâches utilisateur actuellement planifiées pour être exécutées en tant que tâches différées ?
  - a. **atq**
  - b. **atrm**
  - c. **at -c**
  - d. **at --display**
  
- ▶ 2. Quelle commande supprime la tâche utilisateur différée ayant le numéro de travail 5 ?
  - a. **at -c 5**
  - b. **atrm 5**
  - c. **at 5**
  - d. **at --delete 5**
  
- ▶ 3. Quelle commande affiche toutes les tâches utilisateur récurrentes planifiées pour l'utilisateur connecté ?
  - a. **crontab -r**
  - b. **crontab -l**
  - c. **crontab -u**
  - d. **crontab -v**
  
- ▶ 4. Quel format de tâche exécute **/usr/local/bin/daily\_backup** toutes les heures à partir de 9:00 jusqu'à 18:00 tous les jours du lundi au vendredi ?
  - a. **00 \*\*\*Mon-Fri/usr/local/bin/daily\_backup**
  - b. **\* \*/9 \* \* Mon-Fri /usr/local/bin/daily\_backup**
  - c. **00 \*/18 \* \* \* /usr/local/bin/daily\_backup**
  - d. **00 09-18 \* \* Mon-Fri /usr/local/bin/daily\_backup**
  
- ▶ 5. Quel répertoire contient les scripts shell destinés à être exécutés quotidiennement ?
  - a. **/etc/cron.d**
  - b. **/etc/cron.hourly**
  - c. **/etc/cron.daily**
  - d. **/etc/cron.weekly**

► **6. Quel fichier de configuration définit les paramètres des tâches système quotidiennes, hebdomadaires et mensuelles ?**

- a. **/etc/crontab**
- b. **/etc/anacrontab**
- c. **/etc/inittab**
- d. **/etc/sysconfig/crond**

► **7. Quelle unité systemd déclenche régulièrement le nettoyage des fichiers temporaires ?**

- a. **systemd-tmpfiles-clean.timer**
- b. **systemd-tmpfiles-clean.service**
- c. **dnf-makecache.timer**
- d. **unbound-anchor.timer**

## ► SOLUTION

# PLANIFICATION DE TÂCHES À VENIR

Répondez aux questions suivantes en sélectionnant un ou plusieurs éléments.

- ▶ 1. Quelle commande affiche toutes les tâches utilisateur actuellement planifiées pour être exécutées en tant que tâches différées ?
  - a. **atq**
  - b. **atrm**
  - c. **at -c**
  - d. **at --display**
  
- ▶ 2. Quelle commande supprime la tâche utilisateur différée ayant le numéro de travail 5 ?
  - a. **at -c 5**
  - b. **atrm 5**
  - c. **at 5**
  - d. **at --delete 5**
  
- ▶ 3. Quelle commande affiche toutes les tâches utilisateur récurrentes planifiées pour l'utilisateur connecté ?
  - a. **crontab -r**
  - b. **crontab -l**
  - c. **crontab -u**
  - d. **crontab -v**
  
- ▶ 4. Quel format de tâche exécute **/usr/local/bin/daily\_backup** toutes les heures à partir de 9:00 jusqu'à 18:00 tous les jours du lundi au vendredi ?
  - a. **00 \*\*\*Mon-Fri/usr/local/bin/daily\_backup**
  - b. **\* \*/9 \* \* Mon-Fri /usr/local/bin/daily\_backup**
  - c. **00 \*/18 \* \* \* /usr/local/bin/daily\_backup**
  - d. **00 09-18 \* \* Mon-Fri /usr/local/bin/daily\_backup**
  
- ▶ 5. Quel répertoire contient les scripts shell destinés à être exécutés quotidiennement ?
  - a. **/etc/cron.d**
  - b. **/etc/cron.hourly**
  - c. **/etc/cron.daily**
  - d. **/etc/cron.weekly**

► **6. Quel fichier de configuration définit les paramètres des tâches système quotidiennes, hebdomadaires et mensuelles ?**

- a. `/etc/crontab`
- b. `/etc/anacrontab`
- c. `/etc/inittab`
- d. `/etc/sysconfig/crond`

► **7. Quelle unité `systemd` déclenche régulièrement le nettoyage des fichiers temporaires ?**

- a. `systemd-tmpfiles-clean.timer`
- b. `systemd-tmpfiles-clean.service`
- c. `dnf-makecache.timer`
- d. `unbound-anchor.timer`

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Les tâches dont l'exécution est planifiée une fois dans le futur sont appelées tâches ou travaux différés.
- Les tâches utilisateur récurrentes exécutent les tâches utilisateur de manière répétitive.
- Les tâches système récurrentes accomplissent des tâches administratives de manière répétitive qui ont des répercussions sur l'ensemble du système.
- Les unités du minuteur `systemd` peuvent exécuter les tâches différées ou récurrentes.

## CHAPITRE 3

# RÉGLAGE DES PERFORMANCES DU SYSTÈME

### PROJET

Améliorer les performances du système en définissant des paramètres de réglage et en ajustant la priorité d'ordonnancement des processus.

### OBJECTIFS

- Optimiser les performances du système en sélectionnant un profil de réglage géré par le démon tuned.
- Définir et annuler les priorités de processus spécifiques avec les commandes nice et renice.

### SECTIONS

- Ajustement des profils de réglage (et exercice guidé)
- Influence sur l'ordonnancement des processus (et exercice guidé)

### ATELIER

Réglage des performances du système

# AJUSTEMENT DES PROFILS DE RÉGLAGE

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir optimiser les performances du système en sélectionnant un profil de réglage géré par le démon tuned.

## RÉGLAGE DES SYSTÈMES

Les administrateurs système peuvent optimiser les performances d'un système en ajustant divers paramètres de périphérique en fonction de différentes charges de travail. Le démon tuned applique les ajustements de réglage de manière statique et dynamique, en utilisant des profils de réglage reflétant les exigences de la charge de travail.

### Configuration du réglage statique

Le démon tuned applique les paramètres système au démarrage du service ou à la sélection d'un nouveau profil de réglage. Le réglage statique configure les paramètres du noyau prédéfinis dans les profils qui sont appliqués par la commande **tuned** au moment de l'exécution. Avec le réglage statique, les paramètres du noyau sont définis pour des performances globales et ne sont pas ajustés en fonction de l'évolution des niveaux d'activité.

### Configuration du réglage dynamique

Avec le réglage dynamique, le démon tuned surveille l'activité du système et ajuste les paramètres en fonction des changements de comportement à l'exécution. Le réglage dynamique ajuste en continu le réglage en fonction de la charge de travail actuelle, en commençant par les paramètres initiaux déclarés dans le profil de réglage choisi.

Par exemple, les périphériques de stockage sont très utilisés au démarrage et lors de la connexion, mais ont une activité minimale lorsque les charges de travail des utilisateurs consistent à utiliser des navigateurs Web et des clients de messagerie. De même, les processeurs et les périphériques réseau subissent une augmentation d'activité pendant les heures de pointe au cours d'une journée de travail. Le démon tuned surveille l'activité de ces composants. Il ajuste ainsi les paramètres afin d'optimiser les performances lors de périodes de forte activité et de réduire les réglages en cas de faible activité. Le démon tuned utilise les paramètres de performance fournis dans des profils de réglage prédéfinis.

## INSTALLATION ET ACTIVATION DE TUNED

L'installation minimale de Red Hat Enterprise Linux 8 inclut et active le paquetage tuned par défaut. Pour installer et activer le paquetage manuellement :

```
[root@host ~]$ yum install tuned
[root@host ~]$ systemctl enable --now tuned
Created symlink /etc/systemd/system/multi-user.target.wants/tuned.service → /usr/
lib/systemd/system/tuned.service.
```

## SÉLECTION D'UN PROFIL DE RÉGLAGE

L'application tuned fournit des profils divisés selon les catégories suivantes :

- Profils d'économie d'énergie
- Profils d'amélioration des performances

Les profils d'amélioration des performances incluent des profils qui se concentrent sur les aspects suivants :

- Latence faible pour le stockage et le réseau
- Débit élevé pour le stockage et le réseau
- Performances de la machine virtuelle
- Performances de l'hôte de virtualisation

### Réglage des profils distribués avec Red Hat Enterprise Linux 8

PROFIL TUNED	OBJET
équilibré	Idéal pour les systèmes nécessitant un compromis entre économie d'énergie et performances.
desktop	Dérivé du profil balanced. Fournit une réponse plus rapide des applications interactives.
throughput-performance	Réglage du système pour un débit maximal.
latency-performance	Idéal pour les systèmes de serveur nécessitant une faible latence au détriment de la consommation d'énergie.
network-latency	Dérivé du profil latency - performance. Il permet d'ajouter des paramètres de réglage réseau afin de fournir une latence réseau faible.
network-throughput	Dérivé du profil throughput - performance. Des paramètres de réglage réseau supplémentaires sont appliqués pour un débit réseau maximal.
économie d'énergie	Réglage du système pour une économie d'énergie maximale.
oracle	Optimisé pour les charges de base de données Oracle et basé sur le profil throughput - performance.
virtual-guest	Optimise le système pour des performances maximales en cas d'exécution sur une machine virtuelle.
virtual-host	Règle le système pour des performances maximales en cas d'utilisation en tant qu'hôte sur des machines virtuelles.

## GESTION DE PROFILS À PARTIR DE LA LIGNE DE COMMANDE

La commande **tuned-adm** permet de modifier les paramètres d'un démon tuned. La commande **tuned-adm** peut interroger les paramètres actuels, lister les profils disponibles, recommander un profil de réglage pour le système, modifier directement les profils ou désactiver le réglage.

Un administrateur système identifie le profil de réglage actuellement actif avec **tuned-adm active**.

```
[root@host ~]# tuned-adm active  
Current active profile: virtual-guest
```

La commande **tuned-adm list** liste tous les profils de réglage disponibles, y compris les profils intégrés et les profils de réglage personnalisés créés par un administrateur système.

```
[root@host ~]# tuned-adm list  
Available profiles:  
- balanced  
- desktop  
- latency-performance  
- network-latency  
- network-throughput  
- powersave  
- sap  
- throughput-performance  
- virtual-guest  
- virtual-host  
Current active profile: virtual-guest
```

Utilisez **tuned-adm profile *profilename*** pour remplacer le profil actif par un autre profil qui correspond mieux aux exigences de réglage actuelles du système.

```
[root@host ~]$ tuned-adm profile throughput-performance  
[root@host ~]$ tuned-adm active  
Current active profile: throughput-performance
```

La commande **tuned-adm** peut recommander un profil de réglage pour le système. Ce mécanisme est utilisé pour déterminer le profil par défaut d'un système après l'installation.

```
[root@host ~]$ tuned-adm recommend  
virtual-guest
```



### NOTE

La sortie **tuned-adm recommend** est basée sur diverses caractéristiques du système, notamment si le système est une machine virtuelle, et sur d'autres catégories prédéfinies sélectionnées lors de l'installation du système.

Pour annuler les modifications apportées aux paramètres par le profil actuel, basculez vers un autre profil ou désactivez le démon tuned. Désactivez l'activité de réglage tuned avec **tuned-adm off**.

```
[root@host ~]$ tuned-adm off
[root@host ~]$ tuned-adm active
No current active profile.
```

## GESTION DES PROFILS AVEC LA CONSOLE WEB

Pour gérer les profils de performance du système avec la console Web, connectez-vous en tant qu'utilisateur avec privilège. Cliquez l'option Reuse my password for privileged tasks. Cela permet à l'utilisateur d'exécuter des commandes, avec les privilèges sudo, qui modifient les profils de performance du système.

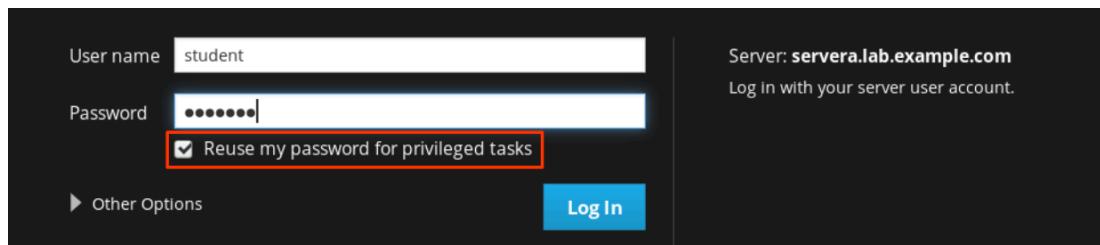


Figure 3.1: Connexion avec privilège à la console Web

En tant qu'utilisateur avec privilège, cliquez sur l'option de menu Systems dans la barre de navigation gauche. Le profil actif actuel est affiché dans le champ Performance Profile. Pour sélectionner un autre profil, cliquez sur le lien du profil actif.

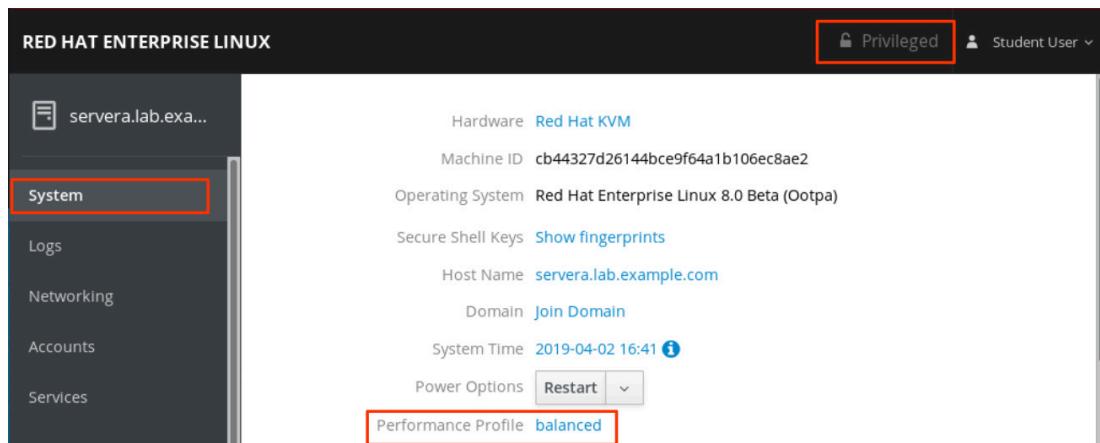


Figure 3.2: Profil de performance actif

Dans l'interface utilisateur Change Performance Profile, faites défiler la liste des profils pour sélectionner celui qui convient le mieux à l'objectif du système.

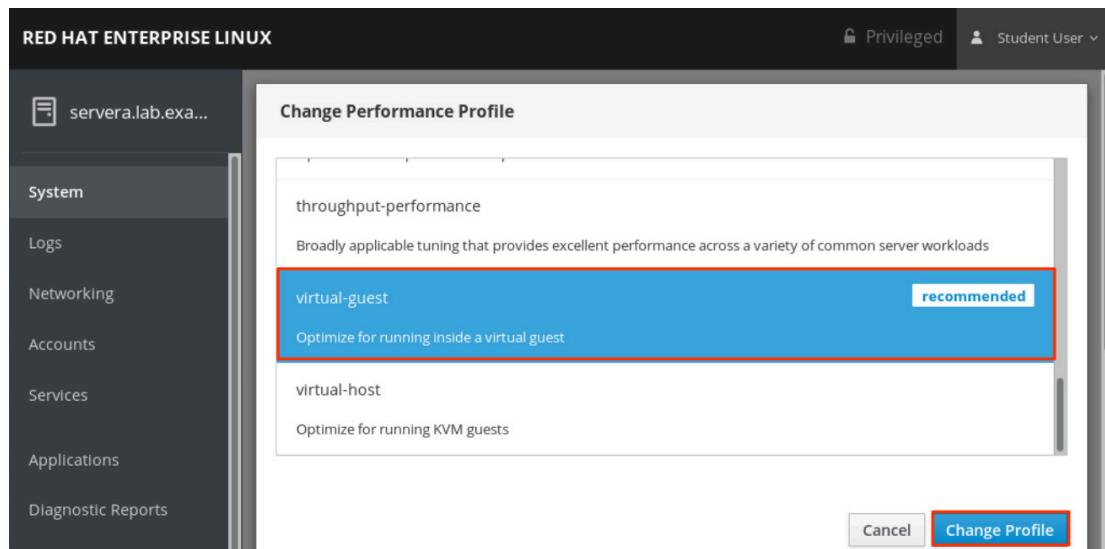


Figure 3.3: Sélection d'un profil de performance préféré

Pour vérifier les modifications, revenez dans la page System principale et contrôlez qu'elle affiche le profil actif dans le champ Performance Profile.

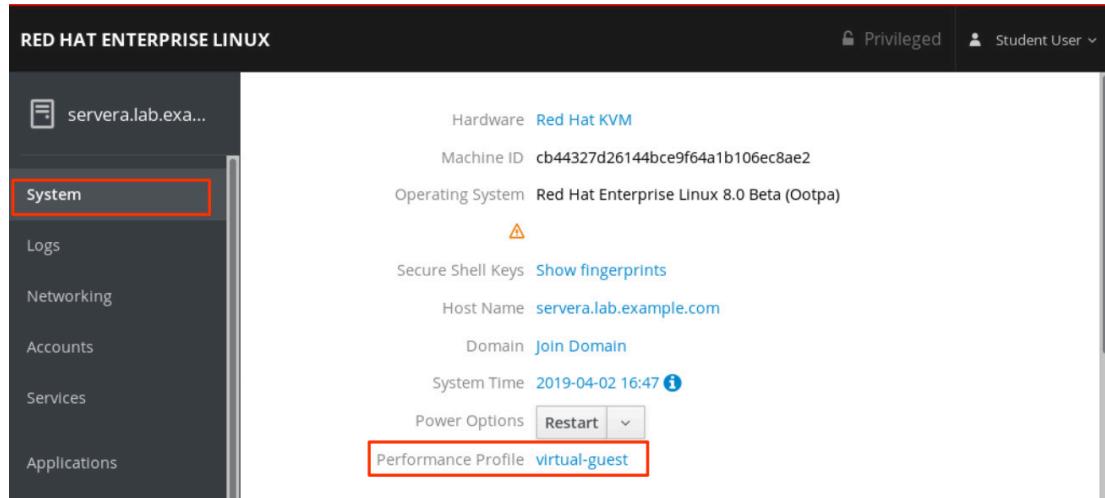


Figure 3.4: Vérification du profil de performance actif



## RÉFÉRENCES

Pages du manuel **tuned(8)**, **tuned.conf(5)**, **tuned-main.conf(5)** et **tuned-adm(1)**

## ► EXERCICE GUIDÉ

# AJUSTEMENT DES PROFILS DE RÉGLAGE

Dans cet exercice, vous allez ajuster les performances d'un serveur en activant le service `tuned` et en appliquant un profil de réglage.

## RÉSULTATS

Vous devez pouvoir configurer un système afin qu'il utilise un profil de réglage.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

À partir de `workstation`, exécutez la commande `lab tuning-profiles start`. La commande exécute un script de démarrage qui détermine si l'hôte `servera` est accessible sur le réseau.

```
[student@workstation ~]$ lab tuning-profiles start
```

- 1. À partir de `workstation`, connectez-vous via SSH à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Vérifiez que le paquetage `tuned` est installé, activé et démarré.

- 2.1. Utilisez `yum` pour vérifier que le paquetage `tuned` est installé.

```
[student@servera ~]$ yum list tuned
...output omitted...
Installed Packages
tuned.noarch                  2.10.0-15.el8          @anaconda
```

- 2.2. La commande `systemctl is-enabled tuned; systemctl is-active tuned` affiche son état d'activation et d'exécution.

```
[student@servera ~]$ systemctl is-enabled tuned; systemctl is-active tuned
enabled
active
```

**CHAPITRE 3 |** Réglage des performances du système

- 3. Listez les profils de réglage disponibles et identifiez le profil actif. Si sudo demande un mot de passe, entrez **student** après l'invite.

```
[student@servera ~]$ sudo tuned-adm list
[sudo] password for student: student
Available profiles:
- balanced           - General non-specialized tuned profile
- desktop            - Optimize for the desktop use-case
- latency-performance - Optimize for deterministic performance at the cost of
                        increased power consumption
- network-latency    - Optimize for deterministic performance at the cost of
                        increased power consumption, focused on low latency
                        network performance
- network-throughput - Optimize for streaming network throughput, generally
                        only necessary on older CPUs or 40G+ networks
- powersave          - Optimize for low power consumption
- throughput-performance - Broadly applicable tuning that provides excellent
                           performance across a variety of common server workloads
- virtual-guest      - Optimize for running inside a virtual guest
- virtual-host        - Optimize for running KVM guests
Current active profile: virtual-guest
```

- 4. Remplacez le profil de réglage actif actuel par **powersave**, puis vérifiez les résultats. Si sudo demande un mot de passe, entrez **student** après l'invite.

4.1. Changez le profil de réglage actif actuel.

```
[student@servera ~]$ sudo tuned-adm profile powersave
```

4.2. Vérifiez que **powersave** est le profil de réglage actif.

```
[student@servera ~]$ sudo tuned-adm active
Current active profile: powersave
```

- 5. Quittez servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez le script **lab tuning-profiles finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab tuning-profiles finish
```

L'exercice guidé est maintenant terminé.

# INFLUENCE SUR L'ORDONNANCEMENT DES PROCESSUS

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir définir et déprioriser les processus spécifiques, avec les commandes **nice** et **renice**.

## ORDONNANCEMENT DES PROCESSUS LINUX ET MULTITÂCHE

Les systèmes informatiques actuels vont des systèmes d'entrée de gamme, dotés d'un processeur unique capable d'exécuter une seule instruction à la fois, aux supercalculateurs haute performance équipés de centaines de processeurs et de douzaines, voire de centaines de coeurs de traitement sur chaque processeur, et qui permettent l'exécution d'un nombre très élevé d'instructions en parallèle. Mais tous ces systèmes présentent un point commun : ils doivent toujours exécuter davantage de threads de processus qu'ils n'ont de processeurs.

Le système Linux et d'autres systèmes d'exploitation exécutent davantage de processus qu'il n'existe d'unités de traitement en utilisant une technique appelée *découpage temporel* (« *time-slicing* ») ou *multitâche*. L'*ordonnanceur de processus* du système d'exploitation alterne rapidement entre les processus sur un même cœur, donnant l'impression que plusieurs processus s'exécutent simultanément.

## PRIORITÉS RELATIVES

Des processus différents présentent des niveaux d'importance divers. Vous pouvez configurer l'ordonnanceur de processus afin qu'il applique des politiques d'ordonnancement différentes aux processus. La politique d'ordonnancement qui régit la plupart des processus exécutés sur un système standard est appelée **SCHED\_OTHER** (qu'on appelle aussi **SCHED\_NORMAL**), mais il existe d'autres politiques en fonction des exigences diverses en matière de charge de travail.

Les processus n'ayant pas tous la même importance, ceux qui s'exécutent selon la politique **SCHED\_NORMAL** peuvent avoir une priorité relative. Cette priorité est appelée *valeur de politesse* d'un processus, et il existe **40** niveaux de politesse différents pour un processus.

Les valeurs de politesse varient entre -20 (priorité la plus haute) et 19 (priorité la plus basse). Par défaut, les processus héritent du niveau de politesse de leur parent qui est généralement 0. Un haut niveau de politesse indique une priorité moindre (le processus laisse ses ressources processeur aux autres), tandis qu'un bas niveau de politesse indique une priorité plus élevée (le processus est moins enclin à abandonner le processeur). S'il n'y a pas de pénurie de ressources (par exemple, quand il y a moins de processus actifs que de coeurs de processeur disponibles), même les processus dotés d'un niveau de politesse élevé continueront à utiliser toutes les ressources processeur possibles. Cependant, quand il y a plus de processus demandeurs de temps processeur que de coeurs disponibles, les processus à haut niveau de politesse reçoivent moins de temps processeur que ceux à bas niveau de politesse.

## DÉFINITION DES NIVEAUX DE POLITESSE ET DES PERMISSIONS

Dans la mesure où l'attribution d'un niveau de politesse faible à un processus gourmand en ressources processeur risque d'impacter négativement les performances des autres processus s'exécutant sur le même système, seul l'utilisateur `root` peut réduire le niveau de politesse d'un processus.

Les utilisateurs sans privilège peuvent uniquement augmenter les niveaux de politesse de leurs propres processus. Ils ne peuvent pas diminuer les niveaux de politesse de leurs processus ni modifier le niveau de politesse des processus des autres utilisateurs.

## AFFICHAGE DES NIVEAUX DE POLITESSE

Plusieurs outils affichent les niveaux de politesse des processus en cours d'exécution. Les outils de gestion de processus, tels que `top`, affichent le niveau de politesse par défaut. D'autres outils tels que la commande `ps` affichent les niveaux de politesse lorsque vous utilisez les options appropriées.

### Affichage des niveaux de politesse avec `top`

Utilisez la commande `top` pour visualiser et gérer les processus de manière interactive. La configuration par défaut affiche deux colonnes intéressantes sur les niveaux de politesse et les priorités. La colonne **NI** affiche la valeur de politesse du processus et la colonne **PR** affiche sa priorité planifiée. Dans l'interface `top`, le niveau de politesse est mis en correspondance avec une file d'attente des priorités système interne, comme indiqué dans le graphique suivant. Par exemple, le niveau de politesse -20 est mis en correspondance avec 0 dans la colonne **PR**. Le niveau de politesse 19 est mis en correspondance avec la priorité 39 dans la colonne **PR**.



Figure 3.5: Niveaux de politesse dans `top`

### Affichage des niveaux de politesse à partir de la ligne de commande

La commande `ps` affiche les niveaux de politesse des processus, mais uniquement en incluant les options de formatage appropriées.

La commande `ps` suivante liste tous les processus avec leur PID, nom, niveau de politesse et classe d'ordonnancement, triés par niveau de politesse décroissant. Les processus dont la colonne de classe d'ordonnancement **CLS** affiche **TS** sont exécutés en vertu de la politique d'ordonnancement **SCHED\_NORMAL**. Les processus dont le niveau de politesse contient un tiret (-) sont exécutés selon d'autres politiques d'ordonnancement et sont interprétés par l'ordonnanceur avec une priorité plus élevée. Les détails des politiques d'ordonnancement supplémentaires ne sont pas abordés dans ce cours.

```
[user@host ~]$ ps axo pid,comm,nice,cls --sort=-nice
  PID COMMAND      NI  CLS
 30 khugepaged    19   TS
```

```
29 ksmd          5  TS
 1 systemd       0  TS
 2 kthreadd      0  TS
 9 ksoftirqd/0   0  TS
10 rcu_sched     0  TS
11 migration/0   -  FF
12 watchdog/0    -  FF
...output omitted...
```

## DÉMARRAGE DES PROCESSUS AVEC DES NIVEAUX DE POLITESSE DIFFÉRENTS

Lors de la création du processus, un processus hérite du niveau de politesse de son parent. Lorsqu'un processus est lancé depuis la ligne de commande, il hérite du niveau de politesse du processus shell à partir duquel il a été démarré. Généralement, ceci se traduit par l'exécution des nouveaux processus avec un niveau de politesse 0.

L'exemple suivant démarre un processus à partir du shell et affiche la valeur de politesse de ce processus. Notez l'utilisation de l'option **PID** dans la commande **ps** pour spécifier la sortie demandée.

```
[user@host ~]$ sha1sum /dev/zero &
[1] 3480
[user@host ~]$ ps -o pid,comm,nice 3480
  PID COMMAND      NI
 3480 sha1sum      0
```

La commande **nice** peut servir à tous les utilisateurs pour démarrer des commandes avec un niveau de politesse par défaut ou supérieur. Si elle ne comporte aucune option, la commande **nice** démarre un processus avec la valeur de politesse par défaut 10.

L'exemple suivant démarre la commande **sha1sum** en tâche de fond avec le niveau de politesse par défaut et affiche le niveau de politesse du processus :

```
[user@host ~]$ nice sha1sum /dev/zero &
[1] 3517
[user@host ~]$ ps -o pid,comm,nice 3517
  PID COMMAND      NI
 3517 sha1sum      10
```

Utilisez l'option **-n** pour appliquer un niveau de politesse défini par l'utilisateur au processus qui est lancé. Par défaut, il convient d'ajouter 10 au niveau de politesse actuel du processus. L'exemple suivant démarre une commande en tâche de fond avec une valeur de politesse définie par l'utilisateur et affiche le niveau de politesse du processus :

```
[user@host ~]$ nice -n 15 sha1sum &
[1] 3521
[user@host ~]$ ps -o pid,comm,nice 3521
  PID COMMAND      NI
 3521 sha1sum      15
```

**IMPORTANT**

Les utilisateurs sans privilège peuvent uniquement augmenter le niveau de politesse à partir de sa valeur actuelle, jusqu'à 19 au maximum. Une fois le niveau augmenté, les utilisateurs sans privilège ne peuvent pas réduire cette valeur pour revenir au niveau de politesse précédent. L'utilisateur **root** peut réduire le niveau de politesse à partir de n'importe quel niveau actuel, jusqu'à la valeur minimale de -20.

## MODIFICATION DU NIVEAU DE POLITESSE D'UN PROCESSUS EXISTANT

Vous pouvez modifier le niveau de politesse d'un processus existant à l'aide de la commande **renice**. Cet exemple utilise l'identificateur PID de l'exemple précédent pour passer du niveau de politesse actuel 15 à 19, qui est le niveau de politesse souhaité.

```
[user@host ~]$ renice -n 19 3521  
3521 (process ID) old priority 15, new priority 19
```

La commande **top** permet également de modifier le niveau de politesse d'un processus. Dans l'interface interactive **top**, appuyez sur l'option **r** pour accéder à la commande **renice**, qui sera suivie du PID à modifier et du nouveau niveau de politesse.

**RÉFÉRENCES**

Pages de manuel **nice(1)**, **renice(1)**, **top(1)** et **sched\_setscheduler(2)**

## ► EXERCICE GUIDÉ

# INFLUENCE SUR L'ORDONNANCEMENT DES PROCESSUS

Dans cet exercice, vous allez ajuster la priorité d'ordonnancement des processus avec les commandes **nice** et **renice**, et observer les effets sur l'exécution des processus.

## RÉSULTATS

Vous devez pouvoir ajuster les priorités d'ordonnancement des processus.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

À partir de **workstation**, exéutez la commande **lab tuning-procscheduling start**. La commande exécute un script de démarrage qui détermine si l'hôte **servera** est accessible sur le réseau.

```
[student@workstation ~]$ lab tuning-procscheduling start
```

- 1. À partir de **workstation**, connectez-vous via SSH à **servera** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Déterminez le nombre de coeurs de processeur sur **servera**, puis démarrez deux instances de la commande **sha1sum /dev/zero** & pour chaque cœur.
- 2.1. Utilisez **grep** pour analyser le nombre de processeurs virtuels existants (coeurs de processeur) du fichier **/proc/cpuinfo**.

```
[student@servera ~]$ grep -c '^processor' /proc/cpuinfo
2
```

- 2.2. Utilisez une commande en boucle pour démarrer plusieurs instances de la commande **sha1sum /dev/zero** &. Démarrer deux instances par processeur virtuel identifié à l'étape précédente. Dans cet exemple, il s'agirait de quatre instances. Les valeurs PID de votre sortie seront différentes de celles de l'exemple.

**CHAPITRE 3 |** Réglage des performances du système

```
[student@servera ~]$ for i in $(seq 1 4); do sha1sum /dev/zero & done  
[1] 2643  
[2] 2644  
[3] 2645  
[4] 2646
```

- 3. Vérifiez que les tâches en arrière-plan sont en cours d'exécution pour chaque processus **sha1sum**.

```
[student@servera ~]$ jobs  
[1]  Running          sha1sum /dev/zero &  
[2]  Running          sha1sum /dev/zero &  
[3]- Running          sha1sum /dev/zero &  
[4]+ Running          sha1sum /dev/zero &
```

- 4. Utilisez les commandes **ps** et **pgrep** pour afficher le pourcentage d'utilisation du processeur pour chaque processus **sha1sum**.

```
[student@servera ~]$ ps u $(pgrep sha1sum)  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
student   2643 49.8  0.0 228360  1744 pts/0      R    11:15   6:09 sha1sum /dev/zero  
student   2644 49.8  0.0 228360  1780 pts/0      R    11:15   6:09 sha1sum /dev/zero  
student   2645 49.8  0.0 228360  1748 pts/0      R    11:15   6:09 sha1sum /dev/zero  
student   2646 49.8  0.0 228360  1780 pts/0      R    11:15   6:09 sha1sum /dev/zero
```

- 5. Arrêtez tous les processus **sha1sum**, puis vérifiez qu'il n'y a aucune tâche en cours d'exécution.

- 5.1. Utilisez la commande **pkill** pour mettre fin à tous les processus en cours d'exécution avec le modèle de nom **sha1sum**.

```
[student@servera ~]$ pkill sha1sum  
[2]  Terminated          sha1sum /dev/zero  
[4]+ Terminated          sha1sum /dev/zero  
[1]- Terminated          sha1sum /dev/zero  
[3]+ Terminated          sha1sum /dev/zero
```

5.2. Vérifiez qu'il n'y a aucune tâche en cours d'exécution.

```
[student@servera ~]$ jobs  
[student@servera ~]$
```

- 6. Démarrez plusieurs instances de **sha1sum /dev/zero &**, puis lancez une instance supplémentaire de **sha1sum /dev/zero &** avec le niveau de politesse 10. Démarrer au moins autant d'instances que le système compte de processeurs virtuels. Dans cet exemple, trois instances standard sont démarrées, plus une autre avec le niveau de politesse le plus élevé.

- 6.1. Utilisez une boucle pour démarrer trois instances de **sha1sum /dev/zero &**.

**CHAPITRE 3 |** Réglage des performances du système

```
[student@servera ~]$ for i in $(seq 1 3); do sha1sum /dev/zero & done  
[1] 1947  
[2] 1948  
[3] 1949
```

- 6.2. Utilisez la commande **nice** pour démarrer la quatrième instance avec le niveau de politesse 10.

```
[student@servera ~]$ nice -n 10 sha1sum /dev/zero &  
[4] 1953
```

- ▶ 7. Utilisez les commandes **ps** et **pgrep** pour afficher le PID, le pourcentage d'utilisation du processeur, la valeur de politesse et le nom d'exécution de chaque processus. L'instance avec la valeur de politesse 10 doit afficher un pourcentage d'utilisation du processeur inférieur à celui des autres instances.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)  
PID %CPU NI COMMAND  
1947 66.0 0 sha1sum  
1948 65.7 0 sha1sum  
1949 66.1 0 sha1sum  
1953 6.7 10 sha1sum
```

- ▶ 8. Utilisez la commande **sudo renice** pour diminuer le niveau de politesse d'un processus de l'étape précédente. Notez la valeur PID de l'instance de processus qui affiche le niveau de politesse 10. Utilisez le PID de ce processus pour diminuer son niveau de politesse à 5.

```
[student@servera ~]$ sudo renice -n 5 1953  
[sudo] password for student: student  
1953 (process ID) old priority 10, new priority 5
```

- ▶ 9. Répétez les commandes **ps** et **pgrep** pour afficher à nouveau le pourcentage d'utilisation du processeur et le niveau de politesse.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)  
PID %CPU NI COMMAND  
1947 63.8 0 sha1sum  
1948 62.8 0 sha1sum  
1949 65.3 0 sha1sum  
1953 9.1 5 sha1sum
```

- ▶ 10. Quittez servera.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez le script **lab tuning-procscheduling finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab tuning-procscheduling finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# RÉGLAGE DES PERFORMANCES DU SYSTÈME

### LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez appliquer un profil de réglage spécifique et ajuster la priorité d'ordonnancement d'un processus existant avec une utilisation élevée du processeur.

### RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Activer un profil de réglage spécifique pour un système informatique.
- Ajuster la priorité d'ordonnancement du processeur d'un processus.

### AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

À partir de workstation, exécutez la commande **lab tuning-review start**. La commande exécute un script de démarrage pour déterminer si l'hôte serverb est accessible sur le réseau.

```
[student@workstation ~]$ lab tuning-review start
```

1. Remplacez le profil de réglage actuel serverb par balanced qui est un profil tuned non spécialisé et général.
2. Deux processus sur serverb consomment un pourcentage élevé d'utilisation du processeur. Ajustez le niveau de politesse de chaque processus à 10 pour allouer plus de temps processeur aux autres processus.

### Évaluation

À partir de workstation, exécutez la commande **lab tuning-review grade** pour confirmer que vous avez réussi cet exercice pratique.

```
[student@workstation ~]$ lab tuning-review grade
```

### Fin

Sur workstation, exécutez le script **lab tuning-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab tuning-review finish
```

L'atelier est maintenant terminé.



## ► SOLUTION

# RÉGLAGE DES PERFORMANCES DU SYSTÈME

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez appliquer un profil de réglage spécifique et ajuster la priorité d'ordonnancement d'un processus existant avec une utilisation élevée du processeur.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Activer un profil de réglage spécifique pour un système informatique.
- Ajuster la priorité d'ordonnancement du processeur d'un processus.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

À partir de workstation, exécutez la commande **lab tuning-review start**. La commande exécute un script de démarrage pour déterminer si l'hôte serverb est accessible sur le réseau.

```
[student@workstation ~]$ lab tuning-review start
```

1. Remplacez le profil de réglage actuel serverb par balanced qui est un profil tuned non spécialisé et général.
  - 1.1. À partir de workstation, ouvrez une session SSH sur serverb en tant que student. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Utilisez **yum** pour vérifier que le paquetage *tuned* est installé.

```
[student@serverb ~]$ yum list tuned
...output omitted...
Installed Packages
tuned.noarch                  2.10.0-15.el8          @anaconda
```

- 1.3. Utilisez la commande **systemctl is-enabled tuned** pour afficher l'état d'activation du service *tuned*.

```
[student@serverb ~]$ systemctl is-enabled tuned  
enabled
```

- 1.4. Listez tous les profils de réglage disponibles et leurs descriptions. Notez que le profil actif actuel est **virtual-guest**.

```
[student@serverb ~]$ sudo tuned-adm list  
[sudo] password for student: student  
Available profiles:  
- balanced - General non-specialized tuned profile  
- desktop - Optimize for the desktop use-case  
- latency-performance - Optimize for deterministic performance at the cost of increased power consumption  
- network-latency - Optimize for deterministic performance at the cost of increased power consumption, focused on low latency network performance  
- network-throughput - Optimize for streaming network throughput, generally only necessary on older CPUs or 40G+ networks  
- powersave - Optimize for low power consumption  
- throughput-performance - Broadly applicable tuning that provides excellent performance across a variety of common server workloads  
- virtual-guest - Optimize for running inside a virtual guest  
- virtual-host - Optimize for running KVM guests  
Current active profile: virtual-guest
```

- 1.5. Remplacez le profil d'activation actif actuel par **balanced**.

```
[student@serverb ~]$ sudo tuned-adm profile balanced
```

- 1.6. Listez les informations récapitulatives du profil tuned actif actuel.

Utilisez la commande **tuned-adm profile\_info** pour vérifier que le profil actif est **balanced**.

```
[student@serverb ~]$ sudo tuned-adm profile_info  
Profile name:  
balanced  
  
Profile summary:  
General non-specialized tuned profile  
...output omitted...
```

2. Deux processus sur **serverb** consomment un pourcentage élevé d'utilisation du processeur. Ajustez le niveau de politesse de chaque processus à 10 pour allouer plus de temps processeur aux autres processus.

- 2.1. Déterminez les deux principaux consommateurs de processeur sur **serverb**. Les principaux consommateurs de processeur sont listés en dernier dans la sortie de la commande. Les valeurs de pourcentage de processeur varient.

```
[student@serverb ~]$ ps aux --sort=pcpu
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
...output omitted...
root       2983  100  0.0 228360  1744 ?          R<   21:08   0:23 md5sum /dev/zero
root       2967  101  0.0 228360  1732 ?          RN   21:08   0:23 sha1sum /dev/zero
[student@serverb ~]$
```

- 2.2. Identifiez le niveau de politesse de chacun des deux principaux consommateurs de processeur.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep md5sum)
PID %CPU NI COMMAND
2967 99.6  2 sha1sum
2983 99.7 -2 md5sum
```

- 2.3. Utilisez la commande **sudo renice -n 10 2967 2983** pour ajuster le niveau de politesse de chaque processus à **10**. Utilisez les valeurs PID identifiées dans la sortie de la commande précédente.

```
[student@serverb ~]$ sudo renice -n 10 2967 2983
[sudo] password for student: student
2967 (process ID) old priority 2, new priority 10
2983 (process ID) old priority -2, new priority 10
```

- 2.4. Vérifiez que le niveau de politesse de chaque processus est 10.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep md5sum)
PID %CPU NI COMMAND
2967 99.6  10 sha1sum
2983 99.7  10 md5sum
```

- 2.5. Quittez serverb.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

## Évaluation

À partir de workstation, exécutez la commande **lab tuning-review grade** pour confirmer que vous avez réussi cet exercice pratique.

```
[student@workstation ~]$ lab tuning-review grade
```

## Fin

Sur workstation, exécutez le script **lab tuning-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab tuning-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Le service `tuned` modifie automatiquement les paramètres du périphérique pour répondre aux besoins spécifiques du système en fonction d'un profil de réglage sélectionné et prédéfini.
- Pour annuler les modifications apportées aux paramètres par un profil sélectionné, basculez vers un autre profil ou désactivez le service `tuned`.
- Le système attribue une priorité relative à un processus afin de déterminer son accès au processeur. Cette priorité s'appelle la valeur de **politesse** d'un processus.
- La commande `nice` attribue une priorité à un processus au démarrage. La commande `renice` modifie la priorité d'un processus en cours d'exécution.



## CHAPITRE 4

# CONTRÔLE DE L'ACCÈS AUX FICHIERS À L'AIDE DES ACL

### PROJET

Interpréter et définir des listes de contrôle d'accès (ACL, Access Control Lists) sur les fichiers pour gérer les situations nécessitant des permissions complexes d'accès pour l'utilisateur et le groupe.

### OBJECTIFS

- Décrire les cas d'utilisation des ACL, identifier les fichiers pour lesquels des ACL sont définies et interpréter les effets de ces ACL.
- Définir et supprimer les ACL sur les fichiers, et définir les ACL par défaut qui sont automatiquement créées par un répertoire sur les fichiers nouvellement créés.

### SECTIONS

- Interprétation des ACL de fichier (et exercice guidé)
- Sécurisation de fichiers à l'aide des ACL (et exercice guidé)

### ATELIER

Contrôle de l'accès aux fichiers à l'aide des ACL

# INTERPRÉTATION DES ACL DE FICHIER

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Décrire les ACL et les différentes options de montage des systèmes de fichiers.
- Visualiser et interpréter les ACL avec **ls** et **getfacl**.
- Décrire le masque ACL et l'ordre de priorité des permissions ACL.
- Identifier où Red Hat Enterprise Linux utilise les ACL par défaut.

## CONCEPTS RELATIFS AUX LISTES DE CONTRÔLE D'ACCÈS

Les permissions de fichiers Linux standard sont satisfaisantes lorsque les fichiers sont utilisés par un seul propriétaire et un seul groupe de personnes désigné. Toutefois, dans certains cas d'utilisation, plusieurs utilisateurs et groupes nommés doivent accéder aux fichiers avec différents ensembles de permissions de fichier. Les *listes de contrôle d'accès (ACL)* offrent cette fonction.

Avec les ACL, vous pouvez accorder des permissions à plusieurs utilisateurs et groupes, identifiés par nom d'utilisateur, nom de groupe, UID ou GID, à l'aide des mêmes indicateurs de permission utilisés avec les autorisations de fichier standard : lecture, écriture et exécution. Ces utilisateurs et groupes supplémentaires, au-delà du propriétaire du fichier et de l'appartenance au groupe du fichier, sont appelés *utilisateurs nommés* et *groupes nommés* respectivement, car ils sont nommés non pas dans une longue liste, mais plutôt dans une liste de contrôle d'accès.

Les utilisateurs peuvent définir des ACL sur les fichiers et les répertoires qu'ils possèdent. Les utilisateurs avec priviléges, affectés à la capacité Linux **CAP\_FOWNER**, peuvent définir des ACL sur n'importe quel fichier ou répertoire. Les nouveaux fichiers et sous-répertoires peuvent hériter automatiquement des paramètres ACL par défaut du répertoire parent, s'ils sont définis. Tout comme les règles d'accès aux fichiers classiques, la hiérarchie de répertoires parents requiert au minimum la permission d'exécution pour *other* afin d'autoriser l'accès aux utilisateurs et groupes nommés.

## Prise en charge des ACL de système de fichiers

Les systèmes de fichiers doivent être montés avec la prise en charge des ACL activée. La prise en charge des ACL est intégrée aux systèmes de fichiers XFS. Les autres systèmes de fichiers, tels que ext3 ou ext4 créés sur Red Hat Enterprise Linux 8, ont l'option **acl** activée par défaut, même si dans les versions antérieures, vous devez confirmer que la prise en charge des ACL est activée. Pour activer la prise en charge des ACL d'un système de fichiers, utilisez l'option **ACL** avec la commande **mount** ou dans l'entrée du système de fichiers dans le fichier de configuration /etc/fstab.

## AFFICHAGE ET INTERPRÉTATION DES PERMISSIONS ACL

La commande **ls -l** ne renvoie qu'un minimum de détails sur les paramètres des ACL :

## CHAPITRE 4 | Contrôle de l'accès aux fichiers à l'aide des ACL

```
[user@host content]$ ls -l reports.txt
-rwxrw----+ 1 user operators 130 Mar 19 23:56 reports.txt
```

Le signe plus (+) à la fin d'une chaîne de permission de 10 caractères indique qu'une structure ACL étendue avec des entrées existe sur ce fichier.

user :

Affiche les paramètres ACL de l'*utilisateur*, qui sont identiques aux paramètres standard du fichier de l'*utilisateur* ; **rwx**.

group :

Affiche les paramètres ACL du *masque* actuel, et non ceux du *groupe propriétaire* ; **rw**.

other :

Affiche les paramètres ACL pour *other*, qui sont identiques aux paramètres de fichiers standard pour *other* (aucun accès).



### IMPORTANT

Modifier les permissions du groupe sur un fichier avec une ACL à l'aide de **chmod** ne modifie pas les permissions du groupe propriétaire. Par contre, cela entraîne la modification du masque ACL. Utilisez **setfac1 -m g::perms file** si votre but est de mettre à jour les permissions du groupe propriétaire du fichier.

## Affichage des ACL d'un fichier

Pour afficher les ACL d'un fichier, utilisez la commande **getfac1 file** :

```
[user@host content]$ getfac1 reports.txt
# file: reports.txt
# owner: user
# group: operators
user::rwx
user:consultant3:---
user:1005:rwx      #effective:rw-
group::rwx        #effective:rw-
group:consultant1:r--
group:2210:rwx    #effective:rw-
mask::rw-
other::---
```

Examinez chaque section de l'exemple précédent :

Entrées commentées :

```
# file: reports.txt
# owner: user
# group: operators
```

Les trois premières lignes sont des commentaires qui identifient le nom du fichier, son propriétaire (**user**) et son groupe propriétaire (**operators**). S'il existe d'autres indicateurs de fichier, tels que **setuid** ou **setgid**, une quatrième ligne de commentaire apparaît pour montrer quels indicateurs sont définis.

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

Entrées des utilisateurs :

```
user::rwx
user:consultant3:---
user:1005:rwx      #effective:rw- ③
```

- ① Permissions du propriétaire du fichier. **user** dispose de permissions **rwx**.
- ② Permissions de l'utilisateur nommé. Une entrée par utilisateur nommé associé à ce fichier. **consultant3** n'a *aucune* permission.
- ③ Permissions de l'utilisateur nommé. L'UID 1005 dispose de permissions **rwx**, mais le masque limite les permissions effectives à **rw**.

Entrées de groupe :

```
group::rwx      #effective:rw- ①
group:consultant1:r-- ②
group:2210:rwx    #effective:rw- ③
```

- ① Permissions du groupe propriétaire. **operators** dispose de permissions **rwx**, mais le masque limite les permissions effectives à **rw**.
- ② Permissions de groupe nommé. Une entrée par groupe nommé associé à ce fichier. **consultant1** dispose uniquement de permission **r**.
- ③ Permissions de groupe nommé. Le GID 2210 dispose de permissions **rwx**, mais le masque les limite à **rw**.

Entrée de masque :

```
mask::rw-
```

Les paramètres de masque indiquent les permissions maximales qui peuvent être accordées à tous les utilisateurs nommés, au groupe propriétaire et aux groupes nommés. L'UID 1005, **operators** et le GID 2210 ne peuvent pas exécuter ce fichier, bien que la permission d'exécution soit valide pour chaque entrée.

Entrée des autres utilisateurs :

```
other::---
```

Permissions pour les autres utilisateurs (ou « world »). Les UID et GID des autres utilisateurs ne bénéficient d'AUCUNE permission.

## Affichage des ACL d'un répertoire

Pour afficher les paramètres ACL d'un répertoire, utilisez la commande **getfacl directory** :

```
[user@host content]$ getfacl .
# file: .
# owner: user
# group: operators
# flags: -s-
user::rwx
user:consultant3:---
user:1005:rwx
```

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

```
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant3:---
default:group::rwx
default:group:consultant1:r-x
default:mask::rwx
default:other::---
```

Examinez chaque section de l'exemple précédent :

Entrées des commentaires du début :

```
# file: .
# owner: user
# group: operators
# flags: -s-
```

Les trois premières lignes sont des commentaires qui identifient le nom du répertoire, son propriétaire (**user**) et son groupe propriétaire (**operators**). S'il existe d'autres indicateurs de fichier (**setuid**, **setgid** ou **sticky**), une quatrième ligne de commentaire indique quels descripteurs sont définis (dans le cas présent, **setgid**).

Entrées ACL standard :

```
user::rwx
user:consultant3:---
user:1005:rwx
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
```

Les permissions ACL relatives à ce répertoire sont identiques à celles du fichier de l'exemple précédent, mais s'appliquent à ce répertoire. La principale différence est l'ajout de la permission d'exécution à ces entrées (le cas échéant) pour activer la permission de recherche dans les répertoires.

Entrées de l'utilisateur par défaut :

```
default:user::rwx
default:user:consultant3:---
```

① ②

- ① Permissions ACL par défaut du propriétaire du fichier. Le groupe propriétaire aura droit à **rwx**, lecture/écriture pour les nouveaux fichiers et exécution pour les nouveaux sous-répertoires.
- ② Permissions ACL par défaut de l'utilisateur nommé. Une entrée par utilisateur nommé, qui se verra attribuer automatiquement les paramètres ACL par défaut appliqués aux

## CHAPITRE 4 | Contrôle de l'accès aux fichiers à l'aide des ACL

nouveaux fichiers ou sous-répertoires. `consultant3` disposera toujours par défaut d'*aucune* permission.

Entrées du groupe par défaut :

```
default:group::rwx  
default:group:consultant1:r-x
```

- ➊ Permissions ACL par défaut du groupe propriétaire. Le groupe propriétaire du fichier aura droit à **rwx**, lecture/écriture pour les nouveaux fichiers et exécution pour les nouveaux sous-répertoires.
- ➋ Permissions ACL par défaut du groupe nommé. Une entrée par groupe nommé qui obtiendra automatiquement les ACL par défaut. `consultant1` obtiendra **rx**, lecture seule pour les nouveaux fichiers et exécution pour les nouveaux sous-répertoires.

Entrée du masque ACL par défaut :

```
default:mask::rwx
```

Les paramètres du masque par défaut indiquent les permissions maximales qui peuvent être initialement accordées pour tout nouveau fichier ou répertoire créé avec des ACL d'utilisateur nommé, de groupe propriétaire ou de groupe nommé : lecture et écriture pour les nouveaux fichiers et exécution pour les nouveaux sous-répertoires. Les nouveaux fichiers ne reçoivent jamais de permission d'exécution.

Entrées « other » par défaut :

```
default:other::---
```

Permissions par défaut pour les *autres* utilisateurs, ou « world ». Les UID et GID des autres ne bénéficient d'*aucune* permission concernant les nouveaux fichiers ou sous-répertoires.

Les entrées de type **default** de l'exemple précédent n'incluent ni l'utilisateur nommé (UID **1005**), ni le groupe nommé (GID **2210**). Par conséquent, ils n'obtiendront pas automatiquement les entrées ACL initiales pour les nouveaux fichiers ou sous-répertoires. Cela permet de limiter efficacement leur accès aux seuls fichiers et sous-répertoires pour lesquels ils disposent déjà d'ACL, ou pour lesquels le propriétaire du fichier ajoute une ACL ultérieurement via la commande **setfac1**. Ils peuvent toujours créer leurs propres fichiers et sous-répertoires.



### NOTE

La sortie de la commande **getfac1** peut être utilisée comme entrée pour **setfac1** afin de restaurer des ACL ou de copier des ACL à partir d'un fichier ou d'un répertoire source et de les enregistrer dans un nouveau fichier. Par exemple, pour restaurer les ACL à partir d'une sauvegarde, utilisez **getfac1 -R /dir1 > file1** pour générer un fichier de vidage de sortie ACL récursif pour le répertoire et son contenu. La sortie peut ensuite être utilisée pour la récupération des ACL originales en passant la sortie sauvegardée comme entrée dans la commande **setfac1**. Par exemple, pour effectuer une mise à jour en bloc du même répertoire dans le chemin actuel, utilisez la commande suivante : **setfac1 --set-file=file1**

## Masque ACL

Le masque ACL définit les permissions maximales qui peuvent être accordées aux *utilisateurs nommés*, au *groupe propriétaire* et aux *groupes nommés*. Il ne limite pas les permissions du *propriétaire du fichier* ou des autres utilisateurs. Tous les fichiers et répertoires qui mettent en œuvre des ACL se verront associer un masque ACL.

Ce masque peut être affiché avec **getfac1** et défini explicitement avec **setfac1**. Il est calculé et ajouté automatiquement s'il n'est pas défini de manière explicite, mais il peut également être hérité d'un paramètre de masque par défaut du répertoire parent. Par défaut, le masque est recalculé à chaque ajout, modification ou suppression d'une des ACL concernées.

## Ordre de priorité des permissions ACL

Lorsque vous déterminez si un processus (un programme en cours d'exécution) peut accéder à un fichier, les permissions de fichier et les ACL sont appliquées comme suit :

- Si le processus est exécuté sous l'identité de l'utilisateur auquel appartient le fichier, les permissions ACL d'utilisateur du fichier s'appliquent.
- Si le processus est exécuté sous l'identité d'un utilisateur répertorié dans une entrée ACL d'utilisateur nommé, les permissions ACL de l'utilisateur nommé s'appliquent (si le masque l'autorise).
- Si le processus est exécuté sous l'identité d'un groupe correspondant au groupe propriétaire du fichier ou d'un groupe doté d'une entrée ACL explicite de groupe nommé, les permissions ACL correspondantes s'appliquent (si le masque l'autorise).
- Sinon, les permissions ACL de type *autre* du fichier s'appliquent.

## EXEMPLES D'UTILISATION DES ACL PAR LE SYSTÈME D'EXPLOITATION

Red Hat Enterprise Linux contient des exemples illustrant l'utilisation classique des ACL dans le cas d'exigences en matière de permissions étendues.

### ACL sur les fichiers journaux systemd

`systemd-journald` utilise les entrées ACL pour permettre un accès en lecture au fichier `/run/log/journal/cb44...8ae2/system.journal` pour les groupes `adm` et `wheel`. Cette ACL permet aux membres des groupes `adm` et `wheel` d'avoir un accès en lecture aux journaux gérés par `journalctl` sans avoir à accorder des permissions spéciales au contenu privilégié à l'intérieur de `/var/log/`, comme `messages`, `secure` ou `audit`.

En raison de la configuration `systemd-journald`, le dossier parent du fichier `system.journal` peut changer, mais `systemd-journald` applique les ACL au nouveau dossier et au nouveau fichier automatiquement.



#### NOTE

Les administrateurs système doivent définir une liste de contrôle d'accès sur le dossier `/var/log/journal/` lorsque `systemd-journald` est configuré pour utiliser le stockage persistant.

```
[user@host ]$ getfacl /run/log/journal/cb44...8ae2/system.journal
getfacl: Removing leading '/' from absolute path names
# file: run/log/journal/cb44...8ae2/system.journal
# owner: root
# group: systemd-journal
user::rw-
group::r--
group:adm:r--
group:wheel:r--
mask::r--
other::---
```

## ACL sur les périphériques gérés systemd

systemd-udev utilise un ensemble de règles udev qui activent la balise uaccess sur certains périphériques, tels que les lecteurs CD/DVD ou graveurs, périphériques de stockage USB, cartes son et bien d'autres. Les règles udev précédemment mentionnées définissent les ACL sur ces périphériques afin de permettre aux utilisateurs connectés à une interface utilisateur graphique (par exemple gdm) d'avoir le contrôle total de ces périphériques.

Les ACL resteront actives jusqu'à ce que l'utilisateur se déconnecte de l'interface utilisateur graphique. Une nouvelle ACL est appliquée au prochain utilisateur qui se connectera à l'interface utilisateur graphique.

Dans l'exemple suivant, vous pouvez voir que user a une entrée ACL avec des permissions **rw** appliquées au périphérique **/dev/sr0** qui est un lecteur CD/DVD.

```
[user@host ]$ getfacl /dev/sr0
getfacl: Removing leading '/' from absolute path names
# file: dev/sr0
# owner: root
# group: cdrom
user::rw-
user:group:rw-
group::rw-
mask::rw-
other::---
```



### RÉFÉRENCES

Pages du manuel **acl(5)**, **getfacl(1)**, **journald.conf(5)**, **ls(1)**, **systemd-journald(8)** et **systemd-udevd(8)**

## ► QUIZ

# INTERPRÉTATION DES ACL DE FICHIER

Reliez les éléments suivants aux éléments correspondants dans le tableau.

default:m::rx /directory

default:user:mary:rx /directory

g::rw /directory

g::rw file

getfacl /directory

group:hug:rwx /directory

user::rx file

user:mary:rx file

DESCRIPTION	OPÉRATION ACL
Afficher la liste de contrôle d'accès sur un répertoire.	
Utilisateur nommé bénéficiant de permissions d'accès à un fichier en lecture et en exécution.	
Propriétaire du fichier bénéficiant de permissions d'accès à un fichier en lecture et en exécution.	
Permissions d'accès en lecture et en écriture à un répertoire, accordées au groupe propriétaire du répertoire.	
Permissions d'accès à un fichier en lecture et en écriture, accordées au groupe propriétaire du fichier.	
Permissions d'accès à un répertoire en lecture, écriture et exécution, accordées à un groupe nommé.	
Permissions d'accès en lecture et en exécution, définies en tant que masque par défaut.	

DESCRIPTION	OPÉRATION ACL
Utilisateur nommé bénéficiant de permissions d'accès initiales aux nouveaux fichiers en lecture, et aux nouveaux sous-répertoires en lecture et en exécution.	

## ► SOLUTION

# INTERPRÉTATION DES ACL DE FICHIER

Reliez les éléments suivants aux éléments correspondants dans le tableau.

DESCRIPTION	OPÉRATION ACL
Afficher la liste de contrôle d'accès sur un répertoire.	<b>getfacl /directory</b>
Utilisateur nommé bénéficiant de permissions d'accès à un fichier en lecture et en exécution.	<b>user:mary:rx file</b>
Propriétaire du fichier bénéficiant de permissions d'accès à un fichier en lecture et en exécution.	<b>user::rx file</b>
Permissions d'accès en lecture et en écriture à un répertoire, accordées au groupe propriétaire du répertoire.	<b>g::rw /directory</b>
Permissions d'accès à un fichier en lecture et en écriture, accordées au groupe propriétaire du fichier.	<b>g::rw file</b>
Permissions d'accès à un répertoire en lecture, écriture et exécution, accordées à un groupe nommé.	<b>group:hug:rwx /directory</b>
Permissions d'accès en lecture et en exécution, définies en tant que masque par défaut.	<b>default:m::rx /directory</b>
Utilisateur nommé bénéficiant de permissions d'accès initiales aux nouveaux fichiers en lecture, et aux nouveaux sous-répertoires en lecture et en exécution.	<b>default:user:mary:rx /directory</b>

# SÉCURISATION DE FICHIERS À L'AIDE DES ACL

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Modifier les permissions ACL normales des fichiers à l'aide de **setfacl**.
- Contrôler les permissions ACL par défaut des nouveaux fichiers et répertoires.

## MODIFICATION DES PERMISSIONS ACL POUR LES FICHIERS

Utilisez **setfacl** pour ajouter, modifier ou supprimer des ACL standard pour les fichiers et les répertoires.

En ce qui concerne les permissions, les listes ACL utilisent le type de représentation normal appliqué par les systèmes de fichiers, à savoir « **r** » pour la lecture, « **w** » pour l'écriture et « **x** » pour l'exécution. Le tiret (« **-** ») signale l'absence de la permission en question. Lors du paramétrage (récuratif) des ACL, vous pouvez utiliser le « **X** » majuscule pour signaler que l'accès en exécution ne doit être appliquée qu'aux répertoires, et non aux fichiers réguliers, sauf si le fichier dispose déjà dudit accès en exécution. Le comportement est le même que celui de **chmod**.

### Ajout ou modification d'ACL

Les ACL peuvent être définies en ligne de commande à l'aide de l'option **-m**, ou transmises via un fichier à l'aide de l'option **-M** (indiquez un tiret « **-** » à la place d'un nom de fichier pour **stdin**). Ces deux options servent à la modification : elles permettent d'ajouter de nouvelles règles à l'ACL ou de remplacer des règles existantes pour un fichier ou un répertoire. Toute autre entrée ACL existante pour le fichier ou le répertoire en question reste en l'état.



#### NOTE

Pour remplacer intégralement les paramètres ACL d'un fichier, utilisez l'option **--set** ou **--set-file**.

Lors de la première définition d'une ACL pour un fichier, si l'opération d'ajout n'inclut aucun paramètre pour les permissions de type *propriétaire du fichier, groupe propriétaire ou autre*, ces dernières sont alors définies sur la base des permissions courantes pour l'accès aux fichiers (également appelées entrées ACL de base, et qui ne peuvent pas être supprimées). De plus, une nouvelle valeur de *masque* est calculée et ajoutée.

Pour ajouter ou modifier une ACL d'*utilisateur* ou d'*utilisateur nommé* :

```
[user@host ~]$ setfacl -m u:name:rX file
```

Si *name* est vide, il s'applique au *propriétaire du fichier*. Dans le cas contraire, la valeur *name* peut correspondre à un nom d'utilisateur ou à un UID. Dans cet exemple, l'accès est autorisé en lecture seule et, si cette permission est déjà accordée, en exécution (sauf si *fichier* est un répertoire, auquel cas il est accessible en exécution afin de permettre au système de le parcourir).

## CHAPITRE 4 | Contrôle de l'accès aux fichiers à l'aide des ACL

Les permissions du *propriétaire du fichier* défini par les ACL sont identiques à celles du *propriétaire du fichier* standard. De ce fait, appliquer un **chmod** aux permissions du *propriétaire du fichier* revient à appliquer **setfac1** aux permissions du *propriétaire du fichier*. **chmod** n'a aucun effet sur les utilisateurs nommés.

Pour ajouter ou modifier une ACL de groupe ou de groupe nommé :

```
[user@host ~]$ setfac1 -m g:name:rw file
```

Le schéma est le même pour l'ajout ou la modification d'une liste ACL d'utilisateur. Si le *nom* est vide, il s'applique au *groupe propriétaire*. Sinon, indiquez un nom de groupe ou un GID pour le *groupe nommé*. Dans cet exemple, l'accès serait autorisé en lecture et en écriture.

**chmod** n'a aucun effet sur les permissions de groupe des fichiers possédant des paramètres ACL, mais permet de mettre à jour le masque ACL.

Pour ajouter ou modifier l'ACL pour *autre* :

```
[user@host ~]$ setfac1 -m o:::- file
```

*autre* n'accepte que les paramètres de permissions. Les paramètres d'autorisation classiques pour les autres sont les suivants : aucune permission, définie avec un tiret (-); et permissions en lecture seule définies comme d'habitude avec **r**. Naturellement, vous pouvez définir n'importe laquelle des permissions standard.

Les permissions ACL pour *autre* et les permissions standard pour *autre* sont équivalentes. De ce fait, appliquer **chmod** aux permissions de type *autre* a le même effet qu'appliquer **setfac1** aux permissions de type *autre*.

Vous pouvez ajouter plusieurs entrées avec la même commande, en utilisant une liste d'entrées séparées par des virgules :

```
[user@host ~]$ setfac1 -m u::rwx,g:consultants:rX,o:::- file
```

Cela permet d'accorder l'accès en lecture, en écriture et en exécution au *propriétaire du fichier*, l'accès en lecture seule et en exécution conditionnelle au groupe **consultants**, mais aucun accès aux utilisateurs de type *autre*. Le *groupe propriétaire* conserve les permissions existantes liées aux ACL ou aux fichiers et les autres entrées « nommées » restent en l'état.

## Utilisation de **getfac1** comme entrée

Vous pouvez utiliser la sortie de **getfac1** comme entrée pour **setfac1**:

```
[user@host ~]$ getfac1 file-A | setfac1 --set-file=- file-B
```

L'option **--set-file** accepte l'entrée d'un fichier ou de *stdin*. Le caractère tiret (-) spécifie l'utilisation de *stdin*. Dans ce cas, *file-B* présente les mêmes paramètres ACL que *file-A*.

## Définition d'un masque ACL explicite

Vous pouvez définir explicitement un masque ACL sur un fichier ou un répertoire pour limiter les permissions effectives maximales accordées aux utilisateurs nommés, au groupe propriétaire et aux groupes nommés. Cela permet de limiter toute permission existante moins restrictive que le masque, sans affecter les permissions moins permissives que ce dernier.

## CHAPITRE 4 | Contrôle de l'accès aux fichiers à l'aide des ACL

```
[user@host ~]$ setfac1 -m m:::r file
```

Cela ajoute une valeur de masque qui n'accorde aux *utilisateurs nommés*, au *groupe propriétaire* et aux *groupes nommés* que l'accès en lecture seule, quels que soient leurs paramètres existants. Les utilisateurs *file owner* (propriétaire de fichier) et *other* ne sont pas affectés par les paramètres du masque.

**getfac1** affiche un commentaire **effective** en regard des entrées limitées par un paramètre du masque.



### IMPORTANT

Par défaut, le masque ACL est recalculé à chaque fois que l'un des paramètres ACL affectés (*utilisateurs nommés*, *groupe propriétaire* ou *groupes nommés*) est modifié ou supprimé, ce qui pourrait redéfinir un précédent paramètre de masque explicite.

Pour éviter que le masque ne soit recalculé, utilisez l'option **-n** ou saisissez un paramètre de masque (**-m m:::perms**) lors de toute opération **setfac1** qui modifie les paramètres ACL affectés par ce masque.

## Modification récursive des ACL

Lorsque vous définissez une ACL sur un répertoire, utilisez l'option **-R** pour appliquer l'ACL de manière récursive. Il convient probablement d'utiliser la permission « **X** » (X majuscule) de manière récursive, afin que les fichiers accessibles en exécution conservent ce paramètre, et que les répertoires soient eux aussi accessibles en exécution, ce qui permet au système de les parcourir. On considère comme une bonne pratique le fait d'utiliser aussi le « **X** » majuscule pour définir des ACL de manière non récursive, car cela empêche les administrateurs d'ajouter involontairement des permissions d'exécution à un fichier standard.

```
[user@host ~]$ setfac1 -R -m u:name:rX directory
```

Cette commande permet d'ajouter le *nom* de l'utilisateur dans le répertoire *directory* et dans tous les sous-répertoires et fichiers existants, ce qui définit l'accès en lecture seule et en exécution sous condition.

## Suppression des ACL

La suppression de règles spécifiques dans une ACL respecte le même format de base que leur modification, à ceci près que « *:perms* » n'est pas spécifié.

```
[user@host ~]$ setfac1 -x u:name,g:name file
```

Cela supprime uniquement le groupe et l'utilisateur nommés dans l'ACL des fichiers ou des répertoires. Toute autre entrée ACL existante reste active.

Vous pouvez utiliser les options de suppression (**-x**) et de modification (**-m**) au cours d'une même opération **setfac1**.

Le masque ne peut être supprimé que lorsqu'aucune autre ACL n'est définie (à part l'ACL de base qui ne peut pas être supprimée). Vous devez donc le supprimer en dernier. Ainsi, le fichier ne comporte plus d'ACL et **ls -l** n'affiche plus le signe « **+** » à côté de la chaîne des permissions.

Pour supprimer toutes les entrées ACL d'un fichier ou d'un répertoire (y compris les ACL *par défaut* pour les répertoires), utilisez la commande suivante :

```
[user@host ~]$ setfacl -b file
```

## CONTRÔLE DES PERMISSIONS ACL PAR DÉFAUT POUR LES FICHIERS

Pour vous assurer que les fichiers et les répertoires créés dans un répertoire héritent de certaines ACL, utilisez l'ACL *par défaut* sur un répertoire. Vous pouvez définir une ACL *par défaut* et chacun des paramètres standard des ACL, y compris un masque par défaut.

Le répertoire lui-même requiert toujours des ACL standard pour le contrôle d'accès, car les ACL *par défaut* ne mettent pas en œuvre le contrôle d'accès pour le répertoire : elles ne prennent en charge que l'héritage des permissions ACL. Par exemple :

```
[user@host ~]$ setfacl -m d:u:name:rx directory
```

Cette commande ajoute un utilisateur nommé par défaut (**d:u:nom**), qui peut accéder aux sous-répertoires en lecture seule et en exécution.

La commande **setfacl** d'ajout d'une ACL *par défaut* pour chaque type d'ACL est exactement la même que pour les ACL standard, mais précédée d'un **d:**. Vous pouvez également utiliser l'option **-d** dans la ligne de commande.



### IMPORTANT

Lorsque vous définissez des ACL *par défaut* pour un répertoire, vérifiez que les utilisateurs pourront accéder au contenu des sous-répertoires créés dans ce dernier, en ajoutant un droit d'exécution dans l'ACL *par défaut*.

Les utilisateurs ne bénéficient pas automatiquement de la permission d'exécution sur les fichiers standard nouvellement créés. En effet, contrairement à celui des nouveaux répertoires, le masque ACL d'un nouveau fichier standard est **rw-**.



### NOTE

L'UID du propriétaire et le GID du groupe principal affectés aux nouveaux fichiers et sous-répertoires restent ceux de l'utilisateur qui les a créés, sauf lorsque le descripteur **setgid** du répertoire parent est activé, auquel cas le GID du groupe principal est identique au GID du répertoire parent.

## Suppression des entrées ACL par défaut

Supprimez une ACL *par défaut* comme vous le faites pour une ACL standard ; là encore, utilisez le préfixe **d:** ou l'option **-d**.

```
[user@host ~]$ setfacl -x d:u:name directory
```

Cette commande supprime l'entrée ACL par défaut ajoutée dans l'exemple précédent.

Pour supprimer toutes les entrée ACL *par défaut* d'un répertoire, exécutez **setfac1 -k directory**.



## RÉFÉRENCES

Pages man **acl(5)**, **setfac1(1)** et **getfac1(1)**

## ► EXERCICE GUIDÉ

# SÉCURISATION DE FICHIERS À L'AIDE DES ACL

Dans cet exercice, vous allez utiliser les entrées ACL pour accorder l'accès à un répertoire d'un groupe, refuser l'accès à un utilisateur, définir l'ACL par défaut sur un répertoire et vérifier que les nouveaux fichiers créés dans ce répertoire héritent de l'ACL par défaut.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Utiliser les entrées ACL pour accorder l'accès à un groupe et refuser l'accès à l'un de ses membres.
- Vérifier que les fichiers et répertoires existants reflètent les nouvelles permissions ACL.
- Définir l'ACL par défaut sur un répertoire et vérifier que les nouveaux fichiers et répertoires héritent de sa configuration.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab acl-secure start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle crée également les utilisateurs, les groupes, les répertoires et les fichiers utilisés dans cet exercice.

```
[student@workstation ~]$ lab acl-secure start
```

Les opérateurs et les consultants sont membres d'une société de support informatique. Ils doivent commencer à partager des informations. `servera` contient un répertoire partagé correctement configuré situé sous `/shares/content` qui héberge des fichiers.

Actuellement, seuls les membres du groupe `operators` ont accès à ce répertoire, mais les membres du groupe `consultants` nécessitent un accès complet à ce répertoire.

L'utilisateur `consultant1` est membre du groupe `consultants`, mais a causé des problèmes à de nombreuses reprises, cet utilisateur ne doit donc pas avoir accès au répertoire.

Votre tâche consiste à ajouter les entrées ACL appropriées au répertoire et à son contenu, de sorte que les membres du groupe `consultants` bénéficient d'un accès complet, mais que l'utilisateur `consultant1` n'en ait aucun. Assurez-vous que les entrées ACL appropriées sont associées aux fichiers et répertoires stockés dans `/shares/content`.

Remarques importantes :

- Les utilisateurs `sysadmin1` et `operator1` sont membres du groupe `operators`.

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

- Les utilisateurs **consultant1** et **consultant2** sont membres du groupe **consultants**.
- Le répertoire **/shares/content** contient un sous-répertoire appelé **server-info** et de nombreux fichiers pour tester l'ACL. De même, le répertoire **/shares/content** contient un script exécutable appelé **loadvg.sh** que vous pouvez utiliser pour les tests.
- Le mot de passe des utilisateurs **sysadmin1**, **operator1**, **consultant1** et **consultant2** est défini sur **redhat**.
- Toutes les modifications doivent s'appliquer uniquement au répertoire **/shares/content** et à ses fichiers ; ne modifiez pas le répertoire **/shares**.

► 1. Connectez-vous à **servera** et basculez vers l'utilisateur **root**.

- 1.1. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

► 2. Ajoutez l'ACL nommée au répertoire **/shares/content** et à l'ensemble de son contenu.

- 2.1. Utilisez la commande **setfacl** pour mettre à jour de manière récursive le répertoire **/shares/content**, afin d'autoriser au groupe **consultants** la lecture, l'écriture et l'exécution conditionnelle.

```
[root@servera ~]# setfacl -Rm g:consultants:rwx /shares/content
```

L'option **-R** signifie récursif, l'option **-m** signifie modifier/ajouter, **rwx** signifie appliquer des permissions d'exécution en lecture, en écriture et en exécution conditionnelle.

- 2.2. Utilisez **setfacl** pour mettre à jour de manière récursive le répertoire **/shares/content**, afin de refuser tout accès à l'utilisateur **consultant1** du groupe **consultants**.

```
[root@servera ~]# setfacl -Rm u:consultant1:- /shares/content
```

L'option **-R** signifie récursif, l'option **-m** signifie modifier/ajouter, **-** signifie ne donner aucun accès.

► 3. Ajoutez l'ACL nommée en tant qu'ACL *par défaut* pour permettre l'ajout d'autres répertoires et fichiers ultérieurement.

- 3.1. Utilisez **setfacl** pour ajouter une règle d'accès par défaut au groupe **consultants**. Autorisez la lecture, l'écriture et l'exécution du répertoire **content**.

```
[root@servera ~]# setfacl -m d:g:consultants:rwx /shares/content
```

L'option **-m** signifie modifier/ajouter ; **d:g** signifie groupe par défaut ; **rwx** signifie permissions de lecture/écriture/exécution (requis pour créer sans problème des sous-répertoires et y accéder).

- 3.2. Utilisez **setfacl** pour ajouter une règle d'accès par défaut pour l'utilisateur **consultant1**. Interdisez tout accès au répertoire **content**.

```
[root@servera ~]# setfacl -m d:u:consultant1:- /shares/content
```

L'option **-m** signifie modifier/ajouter, **d:u** signifie utilisateur par défaut, **-** signifie aucune permission.

- 4. Vérifiez les modifications que vous avez apportées à l'ACL.

**consultant2** doit pouvoir lire n'importe quel fichier et créer un répertoire contenant un nouveau fichier.

**consultant1** ne doit pouvoir ni lire ni écrire des données, ni exécuter aucun fichier. Cela exclut qu'il puisse obtenir la liste du contenu des répertoires.

Utilisez **su - user** pour basculer vers vos utilisateurs de test. Exécutez **exit** ou **Ctrl+D** pour quitter le shell de l'utilisateur de test.

```
[root@servera ~]# exit
[student@servera ~]$ su - consultant2
Password: redhat
[consultant2@servera ~]$ cd /shares/content/
```

- 4.1. Utilisez **cat** pour vérifier que **consultant2** peut lire un fichier.

```
[consultant2@servera content]$ cat serverb-loadavg.txt
#####
serverb.lab.example.com
#####
Wed Mar 25 15:25:19 EDT 2019
#####
ldavg 0.18, 0.06, 0.05
#####
```

- 4.2. Utilisez le script **loadavg.sh** pour vérifier que **consultant2** peut exécuter un fichier.

```
[consultant2@servera content]$ ./loadavg.sh
ldavg 0.00, 0.00, 0.04
```

- 4.3. Créez un répertoire nommé **reports**.

Utilisez **echo** pour créer un fichier avec du contenu, nommez-le **test.txt** et placez-le dans le nouveau répertoire.

Lorsque vous avez terminé, rebasculez vers **student**.

```
[consultant2@servera content]$ mkdir reports
[consultant2@servera content]$ echo "TEST REPORT" > reports/test.txt
[consultant2@servera content]$ exit
logout
[student@servera ~]$
```

- 4.4. Connectez-vous en tant qu'utilisateur **consultant1**. Utilisez **cd** pour essayer d'ouvrir le répertoire sous l'identité de **consultant1**, puis essayez d'exécuter **ls** pour lister le contenu du répertoire. Les deux commandes doivent échouer, et le message **Permission denied** doit s'afficher.

Essayez une ou plusieurs des commandes utilisées par **consultant2**, mais en tant que **consultant1**, pour vérifier l'absence d'accès. Utilisez le chemin complet, **/shares/content**, car vous ne pouvez pas utiliser **cd** pour changer de répertoire.

Rebasculez vers **student** lorsque vous avez terminé vos tests sous l'identité de **consultant1**.

```
[student@servera ~]$ su - consultant1
Password: redhat
[consultant1@servera ~]$ cd /shares/content/
-bash: cd: /shares/content/: Permission denied
[consultant1@servera ~]$ ls /shares/content/
ls: cannot open directory '/shares/content/': Permission denied
[consultant1@servera ~]$ cat /shares/content/serverb-loadavg.txt
cat: /shares/content/serverb-loadavg.txt: Permission denied
[consultant1@servera ~]$ exit
logout
[student@servera ~]$
```

- 4.5. Connectez-vous en tant qu'utilisateur **sysadmin1**. Utilisez **getfacl** pour voir toutes les entrées ACL sur **/shares/content** et les entrées ACL sur **/shares/content/reports**.

Rebasculez vers **student** lorsque vous avez terminé vos tests sous l'identité de **consultant1**.

```
[student@servera ~]$ su - sysadmin1
Password: redhat
[sysadmin1@servera ~]$ getfacl /shares/content
getfacl: Removing leading '/' from absolute path names
# file: shares/content/
# owner: root
# group: operators
# flags: -s-
user::rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
```

```
default:group:consultants:rwx
default:mask::rwx
default:other::---

[sysadmin1@servera ~]$ getfacl /shares/content/reports
getfacl: Removing leading '/' from absolute path names
# file: shares/content/reports
# owner: consultant2
# group: operators
# flags: -S-
user::rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other:---


[sysadmin1@servera ~]$ exit
logout
[student@servera ~]$
```

#### 4.6. Déconnectez-vous de servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez le script **lab acl-secure finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab acl-secure finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# CONTRÔLE DE L'ACCÈS AUX FICHIERS À L'AIDE DES ACL

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez configurer un répertoire de collaboration pour les utilisateurs de deux groupes, en combinant la permission set-GID et les entrées ACL par défaut afin de fournir les permissions d'accès correctes.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Configurer la permission set-GID sur un dossier, pour hériter de la propriété du groupe sur les fichiers et les dossiers qu'il contient.
- Configurer les entrées ACL pour autoriser ou refuser les permissions en lecture/écriture/exécution aux utilisateurs et aux groupes sur les fichiers et les répertoires.
- Configurer la liste de contrôle d'accès par défaut pour obtenir automatiquement les permissions adéquates des ACL et des fichiers sur les nouveaux fichiers et répertoires.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab acl-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle crée également les utilisateurs, les groupes, les répertoires et les fichiers utilisés dans cet exercice.

```
[student@workstation ~]$ lab acl-review start
```

Un organisme de finance boursière met en place un répertoire partagé collaboratif pour y ranger les fichiers sur ses affaires. Les membres du groupe `managers` doivent pouvoir y accéder en lecture et en écriture.

Le cofondateur de l'organisme, `manager1`, a décidé que les membres du groupe `contractors` devaient aussi avoir accès à ce dossier partagé en lecture et en écriture. Cependant, `manager1` ne fait pas confiance à l'utilisateur `contractor3` (un membre du groupe `contractors`), et de ce fait, `contractor3` devrait uniquement avoir accès au répertoire en lecture seule.

`manager1` a créé les utilisateurs et les groupes, et a démarré le processus de configuration du répertoire partagé, en y copiant certains fichiers de modèles. `manager1` étant trop occupé, il vous incombe de terminer cette tâche.

Elle consiste à terminer la configuration du répertoire partagé. Le répertoire et son contenu doivent être attribués au groupe `managers`, et les fichiers rendus accessibles en lecture et en

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

écriture par le propriétaire et le groupe (**managers**). Les autres utilisateurs ne doivent bénéficier d'aucune permission. Vous devez également accorder des permissions de lecture et d'écriture au groupe **contractors**, à l'exception de l'utilisateur **contractor3**, qui n'a qu'une permission de lecture. Assurez-vous que la configuration s'applique aussi bien aux fichiers existants qu'aux fichiers à venir.

Remarques importantes :

- Répertoire partagé : **/shares/cases** sur **serverb**.
  - Les utilisateurs **manager1** et **manager2** sont membres du groupe **managers**.
  - Les utilisateurs **contractor1**, **contractor2** et **contractor3** sont membres du groupe **contractors**.
  - Deux fichiers se trouvent dans le répertoire : **shortlist.txt** et **backlog.txt**.
  - Les cinq utilisateurs ont le même mot de passe : **redhat**.
  - Toutes les modifications doivent s'appliquer uniquement au répertoire **/shares/cases** et à ses fichiers ; ne modifiez pas le répertoire **/shares**.
1. Le répertoire **cases** et son contenu doivent appartenir au groupe **managers**. Les nouveaux fichiers ajoutés au répertoire **cases** doivent être automatiquement affectés au groupe **managers**. Les utilisateurs et les propriétaires de groupe des fichiers existants doivent avoir des permissions de lecture et d'écriture, et les autres utilisateurs ne doivent en avoir aucune.

**NOTE**

(Conseil : n'utilisez pas **setfac1**.)

2. Ajoutez des ACL au répertoire **cases** (et à son contenu) pour accorder aux membres du groupe **contractors** l'accès aux fichiers en lecture/écriture, et au répertoire en exécution. Limitez l'utilisateur **contractor3** à un accès aux fichiers en lecture seule, et au répertoire en exécution.
3. Ajoutez des ACL qui font en sorte que tout nouveau fichier ou répertoire dans le répertoire **cases** se voit appliquer les permissions adéquates pour *tous* les utilisateurs et groupes autorisés.
4. Vérifiez que les modifications apportées au système de fichiers et aux ACL sont correctes. Utilisez **ls** et **getfac1** pour vérifier vos paramètres sur **/shares/cases**.

En tant qu'utilisateur **student**, utilisez **su - user** pour basculer d'abord sur **manager1** et ensuite sur **contractor1**. Vérifiez que vous pouvez modifier un fichier, lire un fichier, créer un répertoire et écrire des données dans un fichier de ce nouveau répertoire. Utilisez **ls** pour vérifier les permissions du nouveau répertoire, et **getfac1** pour vérifier les ACL du nouveau répertoire.

En tant qu'utilisateur **student**, utilisez **su - contractor3** pour changer d'utilisateur. Essayez d'écrire dans un fichier (cela doit échouer) et de créer un répertoire (cela aussi doit échouer). Sous l'identité de l'utilisateur **contractor3**, vous devez pouvoir lire le fichier **shortlist.txt** du répertoire **cases**, ainsi que le fichier de « test » enregistré dans l'un des répertoires créés par les utilisateurs **manager1** et **contractor1**.



### NOTE

La série de tests ci-dessus fait partie des tests que vous pouvez exécuter pour vérifier que les permissions d'accès sont correctes. Vous devez créer des procédures permettant de valider de manière adéquate les accès au sein de votre environnement.

## Évaluation

À partir de **workstation**, exécutez la commande **lab acl-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab acl-review grade
```

## Fin

Sur **workstation**, exécutez la commande **lab acl-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab acl-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# CONTRÔLE DE L'ACCÈS AUX FICHIERS À L'AIDE DES ACL

### LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez configurer un répertoire de collaboration pour les utilisateurs de deux groupes, en combinant la permission set-GID et les entrées ACL par défaut afin de fournir les permissions d'accès correctes.

### RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Configurer la permission set-GID sur un dossier, pour hériter de la propriété du groupe sur les fichiers et les dossiers qu'il contient.
- Configurer les entrées ACL pour autoriser ou refuser les permissions en lecture/écriture/exécution aux utilisateurs et aux groupes sur les fichiers et les répertoires.
- Configurer la liste de contrôle d'accès par défaut pour obtenir automatiquement les permissions adéquates des ACL et des fichiers sur les nouveaux fichiers et répertoires.

### AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab acl-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle crée également les utilisateurs, les groupes, les répertoires et les fichiers utilisés dans cet exercice.

```
[student@workstation ~]$ lab acl-review start
```

Un organisme de finance boursière met en place un répertoire partagé collaboratif pour y ranger les fichiers sur ses affaires. Les membres du groupe `managers` doivent pouvoir y accéder en lecture et en écriture.

Le cofondateur de l'organisme, `manager1`, a décidé que les membres du groupe `contractors` devaient aussi avoir accès à ce dossier partagé en lecture et en écriture. Cependant, `manager1` ne fait pas confiance à l'utilisateur `contractor3` (un membre du groupe `contractors`), et de ce fait, `contractor3` devrait uniquement avoir accès au répertoire en lecture seule.

`manager1` a créé les utilisateurs et les groupes, et a démarré le processus de configuration du répertoire partagé, en y copiant certains fichiers de modèles. `manager1` étant trop occupé, il vous incombe de terminer cette tâche.

Elle consiste à terminer la configuration du répertoire partagé. Le répertoire et son contenu doivent être attribués au groupe `managers`, et les fichiers rendus accessibles en lecture et en

## CHAPITRE 4 | Contrôle de l'accès aux fichiers à l'aide des ACL

écriture par le propriétaire et le groupe (**managers**). Les autres utilisateurs ne doivent bénéficier d'aucune permission. Vous devez également accorder des permissions de lecture et d'écriture au groupe **contractors**, à l'exception de l'utilisateur **contractor3**, qui n'a qu'une permission de lecture. Assurez-vous que la configuration s'applique aussi bien aux fichiers existants qu'aux fichiers à venir.

Remarques importantes :

- Répertoire partagé : **/shares/cases** sur **serverb**.
  - Les utilisateurs **manager1** et **manager2** sont membres du groupe **managers**.
  - Les utilisateurs **contractor1**, **contractor2** et **contractor3** sont membres du groupe **contractors**.
  - Deux fichiers se trouvent dans le répertoire : **shortlist.txt** et **backlog.txt**.
  - Les cinq utilisateurs ont le même mot de passe : **redhat**.
  - Toutes les modifications doivent s'appliquer uniquement au répertoire **/shares/cases** et à ses fichiers ; ne modifiez pas le répertoire **/shares**.
1. Le répertoire **cases** et son contenu doivent appartenir au groupe **managers**. Les nouveaux fichiers ajoutés au répertoire **cases** doivent être automatiquement affectés au groupe **managers**. Les utilisateurs et les propriétaires de groupe des fichiers existants doivent avoir des permissions de lecture et d'écriture, et les autres utilisateurs ne doivent en avoir aucune.



### NOTE

(Conseil : n'utilisez pas **setfac1**.)

- 1.1. Connectez-vous à **serverb** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 1.3. Utilisez la commande **chgrp** pour mettre à jour de manière récursive le groupe propriétaire du répertoire et de son contenu.

```
[root@serverb ~]# chgrp -R managers /shares/cases
```

- 1.4. Utilisez la commande **chmod** pour mettre à jour le descripteur **set-GID** du répertoire.

```
[root@serverb ~]# chmod g+s /shares/cases
```

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

- 1.5. Utilisez **chmod** pour mettre à jour toutes les permissions en **rw** pour le groupe et le propriétaire.

```
[root@serverb ~]# chmod 660 /shares/cases/*
```

2. Ajoutez des ACL au répertoire **cases** (et à son contenu) pour accorder aux membres du groupe **contractors** l'accès aux fichiers en lecture/écriture, et au répertoire en exécution. Limitez l'utilisateur **contractor3** à un accès aux fichiers en lecture seule, et au répertoire en exécution.
- 2.1. Utilisez **setfacl** pour mettre à jour de manière récursive le répertoire **cases** existant et son contenu. Accordez au groupe **contractors** les permissions de lecture, écriture et exécution conditionnelle.

```
[root@serverb ~]# setfacl -Rm g:contractors:rwx /shares/cases
```

- 2.2. Utilisez **setfacl** pour mettre à jour de manière récursive le répertoire **cases** existant et son contenu. Accordez à l'utilisateur **contractor3** les permissions de lecture, d'écriture et d'exécution conditionnelle.

```
[root@serverb ~]# setfacl -Rm u:contractor3:rX /shares/cases
```

3. Ajoutez des ACL qui font en sorte que tout nouveau fichier ou répertoire dans le répertoire **cases** se voit appliquer les permissions adéquates pour *tous* les utilisateurs et groupes autorisés.
- 3.1. Utilisez **setfacl** pour mettre à jour les permissions *par défaut* des membres du groupe **contractors**. Les permissions par défaut sont la lecture, l'écriture et l'exécution (requise pour créer des sous-répertoires et y accéder).

```
[root@serverb ~]# setfacl -m d:g:contractors:rwx /shares/cases
```

- 3.2. Utilisez **setfacl** pour mettre à jour les permissions *par défaut* de l'utilisateur **contractor3**. Les permissions par défaut sont la lecture et l'exécution (requise pour accéder aux sous-répertoires).

```
[root@serverb ~]# setfacl -m d:u:contractor3:rx /shares/cases
```

4. Vérifiez que les modifications apportées au système de fichiers et aux ACL sont correctes. Utilisez **ls** et **getfacl** pour vérifier vos paramètres sur **/shares/cases**.
- En tant qu'utilisateur **student**, utilisez **su - user** pour basculer d'abord sur **manager1** et ensuite sur **contractor1**. Vérifiez que vous pouvez modifier un fichier, lire un fichier, créer un répertoire et écrire des données dans un fichier de ce nouveau répertoire. Utilisez **ls** pour vérifier les permissions du nouveau répertoire, et **getfacl** pour vérifier les ACL du nouveau répertoire.
- En tant qu'utilisateur **student**, utilisez **su - contractor3** pour changer d'utilisateur. Essayez d'écrire dans un fichier (cela doit échouer) et de créer un répertoire (cela aussi doit échouer). Sous l'identité de l'utilisateur **contractor3**, vous devez pouvoir lire le fichier **shortlist.txt** du répertoire **cases**, ainsi que le fichier de « test » enregistré dans l'un des répertoires créés par les utilisateurs **manager1** et **contractor1**.

**CHAPITRE 4 |** Contrôle de l'accès aux fichiers à l'aide des ACL

- 4.1. En tant qu'utilisateur **root**, utilisez **ls** pour vérifier le répertoire **cases** et son contenu. Recherchez les permissions de propriété de groupe, de répertoire et de fichier. Le « **s** » dans les permissions de fichier de groupe indique que le descripteur **set-GID** est défini, et le « **+** » indique que des entrées ACL existent. Lorsque vous avez terminé, quittez la session utilisateur **root**.

```
[root@serverb ~]# ls -ld /shares/cases
drwxrws---+ 2 root managers 46 Mar 29 00:40 /shares/cases
[root@serverb ~]# ls -l /shares/cases
total 8
-rw-rw----+ 1 root managers 44 Mar 29 00:33 backlog.txt
-rw-rw----+ 1 root managers 46 Mar 29 00:33 shortlist.txt
```

- 4.2. Lancez **getfacl** et observez le résultat. Observez le groupe nommé et l'utilisateur nommé dans les entrées des ACL par défaut et standard.

```
[root@serverb ~]# getfacl /shares/cases
# file: shares/cases
# owner: root
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[root@serverb ~]# exit
logout
```

- 4.3. Basculez vers l'utilisateur **manager1** et effectuez les opérations suivantes. Vérifiez que le comportement d'accès obtenu est bien celui que vous attendiez.

```
[student@serverb ~]$ su - manager1
Password: redhat
[manager1@serverb ~]$ cd /shares/cases
[manager1@serverb cases]$ echo hello > manager1.txt
[manager1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[manager1@serverb cases]$ mkdir manager1.dir
[manager1@serverb cases]$ echo hello > manager1.dir/test.txt
[manager1@serverb cases]$ ls -ld manager1.dir
drwxrws---+ 2 manager1 managers 22 Mar 29 00:59 manager1.dir
[manager1@serverb cases]$ ls -l manager1.dir
total 4
-rw-rw----+ 1 manager1 managers 6 Mar 29 00:59 test.txt
```

```
[manager1@serverb cases]$ getfacl manager1.dir
# file: manager1.dir/
# owner: manager1
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[manager1@serverb cases]$ exit
logout
```

- 4.4. Basculez vers l'utilisateur **contractor1** et effectuez les opérations suivantes. Vérifiez que le comportement d'accès obtenu est bien celui que vous attendiez.

```
[student@serverb ~]$ su - contractor1
Password: redhat
[contractor1@serverb ~]$ cd /shares/cases
[contractor1@serverb cases]$ echo hello > manager1.txt
[contractor1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor1@serverb cases]$ mkdir contractor1.dir
[contractor1@serverb cases]$ echo hello > contractor1.dir/test.txt
[contractor1@serverb cases]$ ls -ld contractor1.dir
drwxrws---+ 2 contractor1 managers 22 Mar 29 01:05 contractor1.dir
[contractor1@serverb cases]$ ls -l contractor1.dir
total 4
-rw-rw----+ 1 contractor1 managers 6 Mar 29 01:07 test.txt
[manager1@serverb cases]$ getfacl contractor1.dir
# file: contractor1.dir/
# owner: contractor1
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---
```

```
[contractor1@serverb cases]$ exit  
logout
```

4.5. Basculez vers l'utilisateur **contractor3** et effectuez les opérations suivantes. Vérifiez que le comportement d'accès obtenu est bien celui que vous attendiez.

```
[student@serverb ~]# su - contractor3  
Password: redhat  
[contractor3@serverb ~]# cd /shares/cases  
[contractor3@serverb cases]# echo hello > contractor3.txt  
-bash: contractor3.txt: Permission denied  
[contractor3@serverb cases]# cat shortlist.txt  
###Shortlist of Clients to call###TEMPLATE###  
[contractor3@serverb cases]# mkdir contractor3.dir  
mkdir: cannot create directory 'contractor3.dir': Permission denied  
[contractor3@serverb cases]# cat manager1.dir/test.txt  
hello  
[contractor3@serverb cases]# cat contractor1.dir/test.txt  
hello  
[contractor3@serverb cases]# exit  
logout  
[student@serverb ~]#
```

4.6. Déconnectez-vous de serverb.

```
[student@serverb ~]# exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```



### NOTE

La série de tests ci-dessus fait partie des tests que vous pouvez exécuter pour vérifier que les permissions d'accès sont correctes. Vous devez créer des procédures permettant de valider de manière adéquate les accès au sein de votre environnement.

## Évaluation

À partir de **workstation**, exécutez la commande **lab acl-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab acl-review grade
```

## Fin

Sur **workstation**, exécutez la commande **lab acl-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab acl-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Les ACL donnent un contrôle précis de l'accès aux fichiers et répertoires.
- La commande **getfac1** affiche les ACL d'un fichier ou d'un répertoire.
- La commande **setfac1** définit, modifie et supprime les ACL par défaut et standard des fichiers et des répertoires.
- Utilisez les ACL par défaut pour contrôler les permissions des nouveaux fichiers et répertoires.
- Red Hat Enterprise Linux utilise `systemd` et `udev` pour appliquer des ACL prédéfinies sur des périphériques, des dossiers et des fichiers.

## CHAPITRE 5

# GESTION DE LA SÉCURITÉ AVEC SELINUX

### PROJET

Protéger et gérer la sécurité d'un serveur à l'aide de SELinux.

### OBJECTIFS

- Décrire comment SELinux protège les ressources et comment sélectionner le mode d'exécution.
- Configurer le contexte SELinux d'un fichier pour contrôler la manière dont les processus interagissent avec ce fichier.
- Configurer les valeurs booléennes SELinux pour autoriser les modifications de politique d'exécution selon différents besoins d'accès.
- Examiner les messages du journal SELinux et résoudre les refus AVC SELinux.

### SECTIONS

- Modification du mode d'exécution SELinux (et exercice guidé)
- Contrôle des contextes de fichiers SELinux (et exercice guidé)
- Ajustement de la politique SELinux avec des valeurs booléennes (et exercice guidé)
- Analyse et résolution des problèmes liés à SELinux (et exercice guidé)

### ATELIER

Gestion de la sécurité avec SELinux

# MODIFICATION DU MODE D'EXÉCUTION SELINUX

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Expliquer comment SELinux protège les ressources.
- Modifier le mode SELinux actuel d'un système.
- Définir le mode SELinux par défaut d'un système.

## COMMENT SELINUX PROTÈGE LES RESSOURCES

SELinux fournit un objectif de sécurité essentiel sous Linux, permettant ou interdisant l'accès aux fichiers et à d'autres ressources de façon nettement plus précise que les permissions des utilisateurs.

Les permissions de fichiers déterminent les utilisateurs ou groupes d'utilisateurs qui peuvent accéder à des fichiers spécifiques. Cependant, un utilisateur disposant d'un accès en lecture ou en écriture à un fichier spécifique peut utiliser ce fichier comme bon lui semble, même si cette utilisation ne correspond pas à la manière dont le fichier doit être utilisé.

Par exemple, un accès en écriture défini sur un fichier, tel qu'un fichier de données structuré conçu pour être écrit uniquement à l'aide d'un programme donné, permet à d'autres éditeurs d'ouvrir et de modifier ledit fichier, ce qui peut entraîner une corruption.

Les permissions de fichier ne peuvent pas arrêter ce type d'accès non souhaité. Elles n'ont jamais été conçues pour contrôler *comment* un fichier est utilisé, mais seulement *qui* est autorisé à lire, écrire ou exécuter ce fichier.

SELinux est constitué d'ensembles de politiques, définis par les développeurs d'applications, qui déclarent exactement quelles actions et quels accès sont appropriés et autorisés pour chaque exécutable binaire, fichier de configuration et fichier de données utilisés par une application. Il s'agit d'une *politique ciblée*, car une politique est écrite pour couvrir les activités d'une seule application. Les politiques déclarent des étiquettes prédéfinies qui sont placées sur des programmes, des fichiers et des ports réseau individuels.

## POURQUOI UTILISER SECURITY ENHANCED LINUX ?

Tous les problèmes de sécurité ne peuvent être prédicts à l'avance. SELinux applique un ensemble de règles d'accès empêchant une faiblesse d'une application d'affecter d'autres applications ou le système sous-jacent. SELinux fournit une couche de sécurité supplémentaire ; il ajoute également une couche de complexité qui peut paraître dissuasive aux personnes qui découvrent ce sous-système. L'apprentissage de SELinux peut prendre du temps, mais la politique d'exécution permet qu'une faiblesse dans une partie du système ne se propage pas à d'autres. Si SELinux ne fonctionne pas correctement avec un sous-système particulier, vous pouvez désactiver l'exécution de ce service spécifique jusqu'à ce que vous trouviez une solution au problème sous-jacent.

SELinux comporte trois modes :

- Enforcing : SELinux applique les règles de contrôle d'accès. Les ordinateurs fonctionnent généralement dans ce mode.

- Permissive : SELinux est actif, mais au lieu d'appliquer des règles de contrôle d'accès, il enregistre des avertissements pour des règles qui ont été violées. Ce mode est principalement utilisé pour les tests et la résolution de problèmes.
- Disabled : SELinux est entièrement désactivé : aucune violation SELinux n'est refusée ni même enregistrée. Cette approche est déconseillée !

## CONCEPTS DE BASE RELATIFS À LA SÉCURITÉ SELINUX

Security Enhanced Linux (SELinux) est une couche de sécurité système supplémentaire. La fonction principale de SELinux consiste à protéger les données des utilisateurs contre des services du système dont la sécurité a été compromise. La plupart des administrateurs Linux sont familiarisés avec le modèle de sécurité standard des permissions user/group/other. Il s'agit d'un modèle basé sur des utilisateurs et des groupes, connu en tant que contrôle d'accès discrétionnaire. SELinux fournit une couche de sécurité supplémentaire basée sur les objets et contrôlée par des règles plus compliquées, appelées contrôle d'accès obligatoire.

L'autorisation des accès anonymes distants à un serveur Web requiert l'ouverture de ports sur le pare-feu. Toutefois, cela donne aux personnes malveillantes l'occasion de pirater le système par le biais d'une attaque. Si elles réussissent à compromettre le processus du serveur Web, elles obtiennent ses autorisations. Plus précisément, les permissions de l'utilisateur apache et du groupe apache. Cet utilisateur et ce groupe ont un accès en lecture à la racine du document /var/www/html. Ils ont également accès à /tmp et /var/tmp, ainsi qu'à tous les autres fichiers et répertoires accessibles en écriture par tout le monde.

SELinux est un ensemble de règles de sécurité qui détermine quels processus sont autorisés à accéder à quels fichiers, répertoires et ports. Chaque fichier, processus, répertoire et port dispose d'une étiquette de sécurité spéciale appelée contexte SELinux. Le contexte est un nom qui est utilisé par la politique SELinux pour déterminer si un processus peut ou non accéder à un fichier, un répertoire ou un port. Par défaut, la politique n'autorise aucune interaction, sauf si une règle explicite accorde un accès. S'il n'existe aucune règle d'accès, aucun accès n'est autorisé.

Les étiquettes SELinux comportent plusieurs contextes : **utilisateur**, **rôle**, **type** et **niveau**. Les règles de la politique ciblée, qui constitue la politique par défaut activée dans Red Hat Enterprise Linux, reposent sur le troisième contexte : le contexte de type. Les noms des contextes de types se terminent généralement par **\_t**.

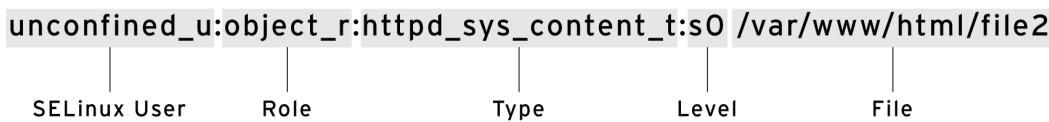


Figure 5.1: Contexte de fichier SELinux

Le contexte de type pour les ports d'un serveur Web est **httpd\_t**. Le contexte de type des fichiers et des répertoires qui se trouvent généralement dans **/var/www/html** est **httpd\_sys\_content\_t**. Le contexte pour les fichiers et répertoires qui se trouvent généralement dans **/tmp** et **/var/tmp** est **tmp\_t**. Le contexte de type pour les ports du serveur Web est **http\_port\_t**.

Le contexte de type d'Apache est **httpd\_t**. Une règle de politique permet à Apache d'accéder aux fichiers et aux répertoires avec le contexte de type **httpd\_sys\_content\_t**. Par défaut, les fichiers qui se trouvent dans **/var/www/html** et d'autres répertoires de serveurs Web ont le contexte de type **httpd\_sys\_content\_t**. La politique ne contient aucune règle d'**accès** pour les fichiers qui se trouvent généralement dans **/tmp** et **/var/tmp**, de sorte que l'accès n'est pas

autorisé. Lorsque SELinux est activé, un utilisateur mal intentionné qui a compromis le processus du serveur WEB ne peut pas accéder au répertoire **/tmp**.

Le serveur MariaDB a le contexte de type **mysqld\_t**. Par défaut, les fichiers qui se trouvent dans **/data/mysql** ont le contexte de type **mysqld\_db\_t**. Ce contexte de type autorise le serveur MariaDB à accéder à ces fichiers, mais désactive l'accès par d'autres services, tels que le service Web Apache.

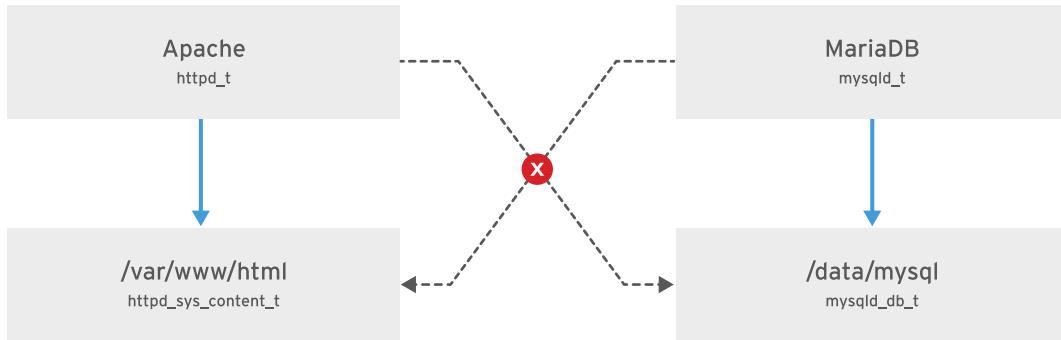


Figure 5.2: Accès SELinux

De nombreuses commandes qui s'appliquent aux fichiers comportent l'option **-Z** pour afficher ou définir le contexte SELinux. Par exemple, **ps**, **ls**, **cp** et **mkdir** utilisent toutes l'option **-Z** pour afficher ou définir le contexte SELinux.

```

[root@host ~]# ps axZ
LABEL PID TTY STAT TIME COMMAND
system_u:system_r:init_t:s0 1 ? Ss 0:09 /usr/lib/systemd/...
system_u:system_r:kernel_t:s0 2 ? S 0:00 [kthreadd]
system_u:system_r:kernel_t:s0 3 ? S 0:00 [ksoftirqd/0]
...output omitted...
[root@host ~]# systemctl start httpd
[root@host ~]# ps -ZC httpd
LABEL PID TTY TIME CMD
system_u:system_r:httpd_t:s0 1608 ? 00:00:05 httpd
system_u:system_r:httpd_t:s0 1609 ? 00:00:00 httpd
...output omitted...
[root@host ~]# ls -Z /home
drwx-----. root root system_u:object_r:lost_found_t:s0 lost+found
drwx-----. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@host ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
  
```

## MODIFICATION DU MODE SELINUX ACTUEL

Le sous-système SELinux fournit des outils pour afficher et modifier le mode. Pour déterminer le mode SELinux actuel, exécutez la commande **getenforce**. Pour définir SELinux dans un autre mode, utilisez la commande **setenforce**:

```
[user@host ~]# getenforce
Enforcing
[user@host ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[user@host ~]# setenforce 0
[user@host ~]# getenforce
Permissive
[user@host ~]# setenforce Enforcing
[user@host ~]# getenforce
Enforcing
```

Vous pouvez également définir le mode SELinux au démarrage en transmettant un paramètre au noyau : l'argument du noyau de **enforcing=0** amorce le système en mode permissive ; la valeur **enforcing=1** définit le mode enforcing. Vous pouvez aussi désactiver complètement SELinux en transmettant le paramètre du noyau **selinux=0**. La valeur **selinux=1** active SELinux.

## DÉFINITION DU MODE SELINUX PAR DÉFAUT

Vous pouvez également configurer SELinux de manière persistante à l'aide du fichier **/etc/selinux/config**. Dans l'exemple ci-dessous (la configuration par défaut), le fichier de configuration définit SELinux sur **enforcing**. Les commentaires montrent également les autres valeurs valides : **permissive** et **disabled**.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.

SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes
#                 are protected.
#       mls - Multi Level Security protection.

SELINUXTYPE=targeted
```

Le système lit ce fichier au démarrage et configure SELinux comme indiqué. Les arguments du noyau (**selinux=0|1** et **enforcing=0|1**) remplacent cette configuration.



### RÉFÉRENCES

Pages du manuel **getenforce(8)**, **setenforce(8)** et **selinux\_config(5)**

## ► EXERCICE GUIDÉ

# MODIFICATION DU MODE D'EXÉCUTION SELINUX

Dans le cadre de cet atelier, vous allez gérer les modes SELinux, à la fois temporairement et de façon persistante.

## RÉSULTATS

Vous devez pouvoir afficher et définir le mode SELinux actuel.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab selinux-opsmode start**. Cette commande exécute un script de démarrage qui détermine si la machine **servera** est accessible sur le réseau.

```
[student@workstation ~]$ lab selinux-opsmode start
```

- ▶ 1. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Redéfinissez le mode SELinux par défaut sur permissive, puis redémarrez.
- 3.1. Utilisez la commande **getenforce** pour vérifier que **servera** est en mode **enforcing**.

```
[root@servera ~]# getenforce
Enforcing
```

- 3.2. Utilisez la commande **vim** pour ouvrir le fichier de configuration **/etc/selinux/config**. Redéfinissez le paramètre **SELINUX** en remplaçant **enforcing** par **permissive**.

```
[root@servera ~]# vim /etc/selinux/config
```

- 3.3. Utilisez la commande **grep** pour vérifier que le paramètre SELINUX est défini sur **permissive**.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=permissive
SELINUXTYPE=targeted
```

- 3.4. Utilisez la commande **systemctl reboot** pour redémarrer servera.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- 4. Le redémarrage de servera prend quelques minutes. Après quelques minutes, connectez-vous à servera en tant qu'utilisateur student. Utilisez la commande **sudo -i** pour devenir l'utilisateur root. Affichez le mode SELinux actuel à l'aide de la commande **getenforce**.

- 4.1. À partir de workstation, utilisez la commande **ssh** pour vous connecter à servera en tant qu'utilisateur student.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 4.2. Utilisez la commande **sudo -i** pour devenir l'utilisateur root.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 4.3. Affichez le mode SELinux actuel à l'aide de la commande **getenforce**.

```
[root@servera ~]# getenforce
Permissive
```

- 5. Dans le fichier **/etc/selinux/config**, redéfinissez le mode SELinux par défaut sur enforcing. Cette modification ne prend effet qu'au prochain redémarrage.

- 5.1. Utilisez la commande **vim** pour ouvrir le fichier de configuration **/etc/selinux/config**. Rétablissez enforcing pour le paramètre SELINUX.

```
[root@servera ~]# vim /etc/selinux/config
```

- 5.2. Utilisez la commande **grep** pour vérifier que le paramètre SELINUX est défini sur **enforcing**.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=enforcing
SELINUXTYPE=targeted
```

- ▶ 6. Utilisez la commande **setenforce** pour définir le mode SELinux actuel sur **enforcing** sans redémarrer. Vérifiez que le mode a bien été défini sur **enforcing** à l'aide de la commande **getenforce**.

```
[root@servera ~]# setenforce 1
[root@servera ~]# getenforce
Enforcing
```

- ▶ 7. Quittez servera.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exécutez le script **lab selinux-opsmode finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab selinux-opsmode finish
```

L'exercice guidé est maintenant terminé.

# CONTÔLE DES CONTEXTES DE FICHIERS SELINUX

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Gérer les règles de la politique SELinux qui déterminent le contexte par défaut pour les fichiers et répertoires à l'aide de la commande **semanage fcontext**.
- Appliquer aux fichiers et répertoires le contexte défini par la politique SELinux à l'aide de la commande **restorecon**.

## CONTEXTE SELINUX INITIAL

Sur les systèmes exécutant SELinux, tous les processus et tous les fichiers sont étiquetés. L'étiquette comporte les informations relatives à la sécurité, appelées contexte SELinux.

En général, les nouveaux fichiers héritent du contexte SELinux du répertoire parent, garantissant ainsi qu'ils ont le contexte approprié.

Mais cette procédure d'héritage peut être compromise de deux manières différentes. Tout d'abord, si vous créez un fichier à un emplacement différent de l'emplacement final prévu, puis déplacez le fichier, celui-ci a toujours le contexte SELinux du répertoire dans lequel il a été créé, et non celui du répertoire de destination. De plus, si vous copiez un fichier en préservant le contexte SELinux, comme avec la commande **cp -a**, le contexte SELinux reflète l'emplacement du fichier d'origine.

L'exemple suivant illustre l'héritage et ses pièges. Considérons ces deux fichiers créés dans **/tmp**, l'un a été déplacé vers **/var/www/html** et l'autre a été copié dans le même répertoire. Notez les contextes SELinux des fichiers. Le fichier qui a été déplacé vers le répertoire **/var/www/html** conserve le contexte de fichier du répertoire **/tmp**. Le fichier qui a été copié dans le répertoire **/var/www/html** a hérité du contexte SELinux du répertoire **/var/www/html**.

La commande **ls -Z** affiche le contexte SELinux d'un fichier. Remarquez l'étiquette du fichier.

```
[root@host ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

De plus, la commande **ls -Zd** affiche le contexte SELinux d'un répertoire :

```
[root@host ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

Notez que **/var/www/html/index.html** a la même étiquette que le répertoire parent **/var/www/html/**. À présent, créez des fichiers en dehors du répertoire **/var/www/html** et remarquez leur contexte de fichier :

```
[root@host ~]# touch /tmp/file1 /tmp/file2
[root@host ~]# ls -Z /tmp/file*
unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```

Déplacez l'un de ces fichiers vers le répertoire **/var/www/html**, copiez-en un autre et remarquez l'étiquette de chacun :

```
[root@host ~]# mv /tmp/file1 /var/www/html/
[root@host ~]# cp /tmp/file2 /var/www/html/
```

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

Le fichier déplacé conserve son étiquette d'origine tandis que le fichier copié hérite de l'étiquette du répertoire **/var/www/html**. **unconfined\_u**: est l'utilisateur, **object\_r**: désigne le rôle et **s0** est le niveau. Un niveau de sensibilité de 0 est le niveau de sensibilité le plus bas possible.

## MODIFICATION DU CONTEXTE SELINUX D'UN FICHIER

Les commandes pour modifier le contexte SELinux des fichiers incluent **semanage fcontext**, **restorecon** et **chcon**.

La méthode recommandée pour définir le contexte SELinux d'un fichier consiste à déclarer l'étiquette par défaut d'un fichier à l'aide de la commande **semanage fcontext** et à appliquer ensuite ce contexte au fichier en utilisant la commande **restorecon**. Cela garantit que l'étiquetage correspond à celui souhaité, même après un réétiquetage complet du système de fichiers.

La commande **chcon** modifie les contextes SELinux. **chcon** définit le contexte de sécurité du fichier qui est stocké dans le système de fichiers. Elle s'avère utile pour effectuer des tests et des expérimentations. Cependant, elle n'enregistre pas les modifications de contexte dans la base de données de contexte SELinux. Lorsqu'une commande **restorecon** s'exécute, les modifications apportées par la commande **chcon** ne sont pas conservées non plus. De plus, si tout le système de fichiers est ré-étiqueté, le contexte SELinux des fichiers modifiés à l'aide de **chcon** est annulé.

L'écran suivant montre un répertoire en cours de création. Le répertoire a une valeur de type de **default\_t**.

```
[root@host ~]# mkdir /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

La commande **chcon** modifie le contexte de fichier du répertoire **/virtual** : la valeur du type devient **httpd\_sys\_content\_t**.

```
[root@host ~]# chcon -t httpd_sys_content_t /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
```

La commande **restorecon** est exécutée et la valeur du type redevient **default\_t**. Remarquez le message **Relabeled**.

```
[root@host ~]# restorecon -v /virtual
Relabeled /virtual from unconfined_u:object_r:object_t:s0 to
unconfined_u:object_r:default_t:s0
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

## DÉFINITION DES RÈGLES DE CONTEXTE DE FICHIER SELINUX PAR DÉFAUT

La commande **semanage fcontext** affiche ou modifie les règles qui sont utilisées par la commande **restorecon** pour définir des contextes de fichiers par défaut. Cette commande utilise les expressions régulières étendues pour spécifier le chemin d'accès et le nom des fichiers. L'expression régulière étendue la plus couramment utilisée dans les règles **fcontext** est **(/.\*)?**, ce qui signifie « un / suivi en option de n'importe quel nombre de caractères ». Cela correspond au répertoire indiqué avant l'expression et à tous les éléments de ce répertoire, de façon récursive.

### Opérations de base sur les contextes de fichiers

Le tableau suivant sert de référence pour les options **semanage fcontext** qui permettent d'ajouter, de supprimer ou de lister les contextes de fichiers SELinux.

#### Commandes semanage fcontext

OPTION	DESCRIPTION
<b>-a, --add</b>	Ajouter une définition du type d'objet spécifié
<b>-d, --delete</b>	Supprimer une définition du type d'objet spécifié
<b>-l, --list</b>	Lister les définitions du type d'objet spécifié

Pour vous assurer de disposer des outils de gestion des contextes SELinux, installez les paquetages **policycoreutil** et **policycoreutil-python** si nécessaire. Ceux-ci contiennent les commandes **restorecon** et **semanage**, respectivement.

Pour vous assurer que tous les fichiers d'un répertoire ont le contexte de fichier correct, exécutez la commande **semanage fcontext -l** suivie de la commande **restorecon**. Dans l'exemple suivant, remarquez le contexte du fichier de chaque fichier avant et après l'exécution des commandes **semanage** et **restorecon**.

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

```
[root@host ~]# semanage fcontext -l
...output omitted...
/var/www(/.*)?    all files    system_u:object_r:httpd_sys_content_t:s0
...output omitted...
```

```
[root@host ~]# restorecon -Rv /var/www/  
Relabeled /var/www/html/file1 from unconfined_u:object_r:user_tmp_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0  
[root@host ~]# ls -Z /var/www/html/file*  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

L'exemple ci-après démontre comment utiliser **semanage** pour ajouter un contexte pour un nouveau répertoire.

```
[root@host ~]# mkdir /virtual  
[root@host ~]# touch /virtual/index.html  
[root@host ~]# ls -Zd /virtual/  
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
```

```
[root@host ~]# ls -Z /virtual/  
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html  
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'  
[root@host ~]# restorecon -RFvv /virtual  
[root@host ~]# ls -Zd /virtual/  
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/  
[root@host ~]# ls -Z /virtual/  
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```



## RÉFÉRENCES

Pages de manuel **chcon(1)**, **restorecon(8)**, **semanage(8)** et **semanage-fcontext(8)**

## ► EXERCICE GUIDÉ

# CONTRÔLE DES CONTEXTES DE FICHIERS SELINUX

Dans le cadre de cet atelier, vous allez apporter une modification persistante au contexte SELinux d'un répertoire et de son contenu.

## RÉSULTATS

Vous devez pouvoir configurer le serveur HTTP *Apache* pour publier le contenu Web à partir d'une racine de documents non standard.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab selinux-filecontexts start**. Cette commande exécute un script de démarrage qui détermine si la machine **servera** est accessible sur le réseau. Elle installe également le service **httpd** et configure le pare-feu sur **servera** afin d'autoriser les connexions HTTP.

```
[student@workstation ~]$ lab selinux-filecontexts start
```

- ▶ 1. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Configurez Apache pour qu'il utilise une racine de documents à un emplacement non standard.

- 3.1. Créez la racine de documents **/custom** à l'aide de la commande **mkdir**.

```
[root@servera ~]# mkdir /custom
```

**CHAPITRE 5 |** Gestion de la sécurité avec SELinux

- 3.2. Créez le fichier **index.html** à la racine de documents **/custom** à l'aide de la commande **echo**.

```
[root@servera ~]# echo 'This is SERVERA.' > /custom/index.html
```

- 3.3. Configurez Apache pour qu'il utilise le nouvel emplacement de racine de documents. Vous devez remplacer les deux occurrences de **/var/www/html** par **/custom** dans le fichier de configuration d'Apache **/etc/httpd/conf/httpd.conf**.

```
[root@servera ~]# vim /etc/httpd/conf/httpd.conf
[root@servera ~]# grep custom /etc/httpd/conf/httpd.conf
DocumentRoot "/custom"
<Directory "/custom">
```

- 4. Démarrez et activez le service Web Apache et vérifiez que le service est en cours d'exécution.

- 4.1. Démarrez et activez le service Web Apache à l'aide de la commande **systemctl**.

```
[root@servera ~]# systemctl enable --now httpd
```

- 4.2. Utilisez la commande **systemctl** pour vérifier que le service est en cours d'exécution.

```
[root@servera ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2019-03-25 19:16:48 CET; 15h ago
    Docs: man:httpd.service(8)
 Main PID: 6565 (httpd)
   Status: "Total requests: 16; Idle/Busy workers 100/0;Requests/sec: 0.000285;
 Bytes served/sec: 0 B/sec"
     Tasks: 213 (limit: 11406)
   Memory: 37.3M
    CGroup: /system.slice/httpd.service
            ├─6565 /usr/sbin/httpd -DFOREGROUND
            ├─6566 /usr/sbin/httpd -DFOREGROUND
            ├─6567 /usr/sbin/httpd -DFOREGROUND
            ├─6568 /usr/sbin/httpd -DFOREGROUND
            └─6569 /usr/sbin/httpd -DFOREGROUND

Mar 25 19:16:48 servera.lab.example.com systemd[1]: Starting The Apache HTTP
Server...
Mar 25 19:16:48 servera.lab.example.com httpd[6565]: Server configured, listening
on: port 80
Mar 25 19:16:48 servera.lab.example.com systemd[1]: Started The Apache HTTP
Server.
```

- 5. Ouvrez un navigateur Web sur **workstation** et tentez d'accéder à **http://servera/index.html**. Vous obtenez un message d'erreur indiquant que vous n'êtes pas autorisé à accéder au fichier.

- ▶ 6. Pour autoriser l'accès au fichier **index.html** sur **servera**, vous devez configurer SELinux. Créez une règle de contexte de fichier SELinux définissant le contexte de type **httpd\_sys\_content\_t** pour le répertoire **/custom** et pour tous les fichiers situés en aval.

```
[root@servera ~]# semanage fcontext -a -t httpd_sys_content_t '/custom(/.*)?'
```

- ▶ 7. Utilisez la commande **restorecon** pour modifier le contexte de fichier.

```
[root@servera ~]# restorecon -Rv /custom
Relabeled /custom from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
```

- ▶ 8. Tentez d'accéder à nouveau `http://servera/index.html`. Vous devez recevoir le message **This is SERVERA**.
- ▶ 9. Quittez **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur **workstation**, exécutez le script **lab\_selinux-filecontexts\_finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab_selinux-filecontexts finish
```

L'exercice guidé est maintenant terminé.

# AJUSTEMENT DE LA POLITIQUE SELINUX AVEC DES VALEURS BOOLEENNES

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Activer et désactiver les règles de la politique SELinux à l'aide de **setsebool**.
- Gérer la valeur persistante des valeurs booléennes SELinux en utilisant la commande **semanage boolean -l**.
- Consulter les pages de manuel qui se terminent par **\_selinux** afin d'y trouver des informations utiles sur les valeurs booléennes SELinux.

## VALEURS BOOLEENNES SELINUX

Les valeurs booléennes SELinux sont des commutateurs qui modifient le comportement de la politique SELinux. Les valeurs booléennes SELinux sont des règles pouvant être activées ou désactivées. Les administrateurs chargés de la sécurité peuvent les utiliser pour affiner la politique afin de procéder à des ajustements sélectifs.

Les pages de manuel SELinux, fournies avec le paquetage *selinux-policy-doc*, décrivent l'objet des valeurs booléennes disponibles. La commande **man -k '\_selinux'** liste ces pages de manuel.

Les commandes utiles pour gérer les valeurs booléennes SELinux incluent **getsebool**, qui liste les valeurs booléennes et leur état, et **setsebool** qui modifie ces valeurs booléennes. **setsebool -P** modifie la politique SELinux afin de rendre la modification persistante. La commande **semanage boolean -l** indique si une valeur booléenne est ou non persistante, et permet d'afficher une brève description de celle-ci.

Les utilisateurs sans privilège peuvent exécuter la commande **getsebool**, mais vous devez être un superutilisateur pour exécuter **semanage boolean -l** et **setsebool -P**.

```
[user@host ~]$ getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
...output omitted...
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

```
[user@host ~]$ setsebool httpd_enable_homedirs on
Could not change active booleans. Please try as root: Permission denied
[user@host ~]$ sudo setsebool httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , off)  Allow httpd to enable homedirs
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

L'option **-P** écrit toutes les valeurs en attente dans la politique, les rendant ainsi persistantes lors des redémarrages. Dans l'exemple qui suit, notez les valeurs entre parenthèses : les deux sont maintenant définies sur **on**.

```
[user@host ~]$ setsebool -P httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , on)  Allow httpd to enable homedirs
```

Pour lister valeurs booléennes dans lesquelles l'état actuel diffère de l'état par défaut, exécutez **semanage boolean -l -C**.

```
[user@host ~]$ sudo semanage boolean -l -C
SELinux boolean           State  Default Description
cron_can_relabel          (off , on)  Allow cron to can relabel
```



## RÉFÉRENCES

Pages du manuel **booleans(8)**, **getsebool(8)**, **setsebool(8)**, **semanage(8)**, **semanage-boolean(8)**

## ► EXERCICE GUIDÉ

# AJUSTEMENT DE LA POLITIQUE SELINUX AVEC DES VALEURS BOOLÉENNES

Apache peut publier le contenu Web hébergé dans le répertoire personnel des utilisateurs, mais SELinux l'en empêche par défaut. Dans le cadre de cet exercice, vous allez identifier et modifier la valeur booléenne SELinux qui permet à Apache d'accéder au répertoire personnel des utilisateurs.

## RÉSULTATS

Vous devez pouvoir configurer Apache de sorte qu'il publie du contenu Web à partir des répertoires personnels des utilisateurs.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab selinux-booleans start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle installe également le service `httpd` et configure le pare-feu sur `servera` afin d'autoriser les connexions HTTP.

```
[student@workstation ~]$ lab selinux-booleans start
```

- ▶ 1. Utilisez la commande `ssh` pour vous connecter à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande `sudo -i` pour basculer vers l'utilisateur `root`. Le mot de passe de l'utilisateur `student` est `student`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Pour activer la fonctionnalité Apache qui permet aux utilisateurs de publier du contenu Web depuis leur répertoire personnel, vous devez éditer le fichier de configuration `/etc/httpd/conf.d/userdir.conf`. Commentez la ligne qui définit `UserDir` sur `disabled` et supprimez le commentaire de la ligne qui définit `UserDir` sur `public_html`.

```
[root@servera ~]# vim /etc/httpd/conf.d/userdir.conf
#UserDir disabled
UserDir public_html
```

- 4. Utilisez la commande **grep** pour confirmer les modifications.

```
[root@servera ~]# grep '#UserDir' /etc/httpd/conf.d/userdir.conf
#UserDir disabled
[root@servera ~]# grep '^ *UserDir' /etc/httpd/conf.d/userdir.conf
UserDir public_html
```

- 5. Redémarrez et activez le service Web Apache pour que vos modifications soient prises en compte.

```
[root@servera ~]# systemctl enable --now httpd
```

- 6. Dans une autre fenêtre de terminal, connectez-vous en tant que **student**. Utilisez la commande SSH pour vous connecter au **servera**. Créez un contenu Web à publier à partir du répertoire personnel d'un utilisateur.

6.1. Dans une autre fenêtre de terminal, connectez-vous en tant que **student**. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

6.2. Utilisez la commande **mkdir** pour créer un répertoire appelé **~/public\_html**.

```
[student@servera ~]$ mkdir ~/public_html
```

6.3. Créez le fichier **index.html** avec le contenu suivant :

```
[student@servera ~]$ echo 'This is student content on SERVERA.' > \
~/public_html/index.html
```

6.4. Utilisez la commande **chmod** pour modifier les permissions du répertoire personnel de **student** de sorte qu'Apache puisse accéder au sous-répertoire **public\_html**.

```
[student@servera ~]$ chmod 711 ~
```

- 7. Ouvrez un navigateur Web sur **workstation** et tentez d'accéder à l'URL suivante : **http://servera/~student/index.html**. Vous obtenez un message d'erreur indiquant que vous n'êtes pas autorisé à accéder au fichier.
- 8. Depuis la fenêtre de terminal, avec un accès **root**, utilisez la commande **getsebool** pour déterminer si des valeurs booléennes limitent l'accès aux répertoires personnels.

```
[root@servera ~]# getsebool -a | grep home
...output omitted...
httpd_enable_homedirs --> off
...output omitted...
```

- ▶ 9. Dans la fenêtre de terminal, avec un accès **root**, utilisez la commande **setsebool** pour permettre l'accès au répertoire personnel de manière persistante.

```
[root@servera ~]# setsebool -P httpd_enable_homedirs on
```

- ▶ 10. Tentez d'accéder à nouveau `http://servera/~student/index.html`. Vous devez obtenir le message : **This is student content on SERVERA**.

- ▶ 11. Quittez servera.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exécutez le script **lab\_selinux-booleans finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab_selinux-booleans finish
```

L'exercice guidé est maintenant terminé.

# ANALYSE ET RÉSOLUTION DES PROBLÈMES LIÉS À SELINUX

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Utiliser les outils d'analyse des journaux SELinux.
- Afficher des informations utiles lors de la résolution de problèmes liés à SELinux à l'aide de la commande **sealert**.

## RÉSOLUTION DES PROBLÈMES LIÉS À SELINUX

Il est important de comprendre quelles actions vous devez effectuer lorsque SELinux empêche l'accès aux fichiers sur un serveur censé être accessible. Utilisez les étapes suivantes pour vous aider à résoudre ces problèmes :

1. Avant de procéder à des ajustements, envisagez que SELinux fait peut-être son travail correctement en interdisant la tentative d'accès. Si un serveur Web tente d'accéder aux fichiers de **/home**, cela peut être un signe que le service a été compromis si le contenu Web n'est pas publié par les utilisateurs. Si l'accès aurait dû être accordé, il convient de prendre d'autres mesures pour résoudre le problème.
2. Le problème SELinux le plus courant est un contexte de fichier incorrect. Ce problème peut survenir lorsqu'un fichier est créé à un emplacement avec un contexte de fichier spécifique, puis déplacé dans un endroit où un autre contexte est en vigueur. Dans la plupart des cas, l'exécution de **restorecon** suffit à corriger le problème. La résolution des problèmes par ce biais n'a qu'une incidence mineure sur la sécurité du reste du système.
3. Un autre remède à un accès trop restrictif peut être l'ajustement d'une valeur booléenne. Par exemple, la valeur booléenne **ftpd\_anon\_write** détermine si les utilisateurs FTP anonymes peuvent télécharger des fichiers. Vous devez activer cette valeur booléenne pour permettre aux utilisateurs FTP anonymes de télécharger des fichiers sur un serveur. L'ajustement des valeurs booléennes demande davantage de précautions car elles peuvent avoir un impact important sur la sécurité du système.
4. Il est possible que la politique SELinux comporte un bogue qui bloque un accès légitime. SELinux ayant beaucoup évolué, ce type de problème est rare. Si un bogue de la politique est clairement identifié, signalez-le à l'assistance technique de Red Hat afin qu'il puisse être résolu.

## CONTRÔLE DES VIOLATIONS SELINUX

Installez le paquetage **setroubleshoot-server** pour envoyer les messages SELinux dans **/var/log/messages**. **setroubleshoot-server** écoute les messages d'audit de **/var/log/audit/audit.log** et envoie un court récapitulatif dans **/var/log/messages**. Ce résumé comprend les *identificateurs uniques (UUID)* des violations SELinux. Ces identificateurs sont utiles pour recueillir des informations complémentaires. La commande **sealert -l *UUID*** permet de produire un rapport pour un incident spécifique. Utilisez **sealert -a /var/log/audit/audit.log** pour produire un rapport sur tous les incidents de ce fichier.

Examinez l'exemple suivant qui représente une séquence de commandes sur un serveur Web Apache :

```
[root@host ~]# touch /root/file3
[root@host ~]# mv /root/file3 /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

Vous attendez du serveur Web qu'il fournisse le contenu de **file3** mais à la place, il renvoie une erreur **permission denied**. Le fait d'examiner à la fois **/var/log/audit/audit.log** et **/var/log/messages** permet de révéler quelques informations complémentaires à propos de cette erreur.

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1392944135.482:429): avc: denied { setattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...output omitted...
[root@host ~]# tail /var/log/messages
...output omitted...
Feb 20 19:55:42 host setroubleshoot: SELinux is preventing /usr/sbin/httpd
from getattr access on the file . For complete SELinux messages. run
sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

Ces deux fichiers journaux indiquent que le coupable est un refus SELinux. La commande **sealert** qui fait partie de la sortie dans **/var/log/messages** fournit quelques informations complémentaires, notamment un correctif éventuel.

```
[root@host ~]# sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
SELinux is preventing /usr/sbin/httpd from getattr access on the file .

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed setattr access on the
file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:admin_home_t:s0
```

```

Target Objects          [ file ]
Source                 httpd
Source Path            /usr/sbin/httpd
Port                  <Unknown>
Host                  servera
Source RPM Packages   httpd-2.4.6-14.el7.x86_64
Target RPM Packages   selinux-policy-3.12.1-124.el7.noarch
Policy RPM             selinux-policy-3.12.1-124.el7.noarch
Selinux Enabled        True
Policy Type            targeted
Enforcing Mode        Enforcing
Host Name              servera
Platform               Linux servera 3.10.0-84.el7.x86_64 #1
                        SMP Tue Feb 4 16:28:19 EST 2014 x86_64 x86_64
Alert Count            2
First Seen             2014-02-20 19:55:35 EST
Last Seen              2014-02-20 19:55:35 EST
Local ID               613ca624-248d-48a2-a7d9-d28f5bbe2763

Raw Audit Messages
type=AVC msg=audit(1392944135.482:429): avc: denied { getattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file

type=SYSCALL msg=audit(1392944135.482:429): arch=x86_64 syscall=lstat
success=no exit=EACCES a0=7f9fed0edeaa8 a1=7fff7bffc770 a2=7fff7bffc770
a3=0 items=0 ppid=1608 pid=1609 auid=4294967295 uid=48 gid=48 euid=48
suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295
comm=httpd exe=/usr/sbin/httpd subj=system_u:system_r:httpd_t:s0 key=(null)

Hash: httpd,httpd_t,admin_home_t,file,getattr

```



### NOTE

La section **Raw Audit Messages** indique le fichier cible qui pose problème, **/var/www/html/file3**. En outre, le contexte cible, **tcontext**, semble être sans rapport avec un serveur Web. Utilisez la commande **restorecon /var/www/html/file3** pour corriger le contexte de fichier. Si d'autres fichiers nécessitent des ajustements, **restorecon** peut restaurer le contexte de façon récursive : **restorecon -R /var/www/**.

La section **Raw Audit Messages** de la commande **sealert** contient des informations de **/var/log/audit.log**. Pour rechercher le fichier **/var/log/audit.log**, utilisez la commande **ausearch**. L'option **-m** effectue une recherche sur le type de message. L'option **-ts** effectue une recherche fondée sur le temps.

**CHAPITRE 5 |** Gestion de la sécurité avec SELinux

```
[root@host ~]# ausearch -m AVC -ts recent
-----
time->Tue Apr  9 13:13:07 2019
type=PROCTITLE msg=audit(1554808387.778:4002):
  proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1554808387.778:4002): arch=c000003e syscall=49
  success=no exit=-13 a0=3 a1=55620b8c9280 a2=10 a3=7ffed967661c items=0
  ppid=1 pid=9340 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
  sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
  subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1554808387.778:4002): avc:  denied  { name_bind }
for  pid=9340 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
```

## CONSOLE WEB

Si la *console Web* est installée, elle peut également être utilisée pour résoudre les problèmes liés à SELinux. Connectez-vous à la console Web et sélectionnez SELinux dans le menu de gauche. La fenêtre SELinux Policy vous informe de la politique d'exécution actuelle. Tous les problèmes sont détaillés dans la section SELinux Access Control Errors.

### SELinux Policy

Enforce policy: **ON**

### SELinux Access Control Errors

> SELinux is preventing /usr/sbin/httpd from open access on the file /lab-content/lab.html.

**Figure 5.3: Politique SELinux dans la console Web**

Cliquez sur le caractère > pour afficher les détails de l'erreur. Cliquez sur solution details pour montrer tous les détails et la solution possibles.

### SELinux Policy

Enforce policy: **ON**

### SELinux Access Control Errors

> SELinux is preventing /usr/sbin/httpd from open access on the file /lab-content/lab.html.

Solutions	Audit log	Occurred between Last Thursday at 11:22 AM and Last Thursday at 3:08 PM	
<p>If you believe that httpd should be allowed open access on the lab.html file by default.            You should report this as a bug. You can generate a local policy module to allow this access.  <a href="#">solution details</a></p> <p>Allow this access for now by executing: # ausearch -c 'httpd' --raw   audit2allow -M my-httpd # semodule -X 300 -l my-httpd.pp</p>		10	

**Figure 5.4: Solution relative à la politique SELinux dans la console Web**

Une fois le problème résolu, la section SELinux Access Control Errors ne doit plus afficher l'erreur. Si le message **No SELinux alerts** apparaît, alors tous les problèmes ont été corrigés.

## SELinux Policy

Enforce policy: **ON**

## SELinux Access Control Errors

No SELinux alerts.

**Figure 5.5: Aucune alerte SELinux dans la console Web**



### RÉFÉRENCES

Page du manuel (8)**sealert**

## ► EXERCICE GUIDÉ

# ANALYSE ET RÉSOLUTION DES PROBLÈMES LIÉS À SELINUX

Dans le cadre de cet atelier, vous allez apprendre à résoudre les problèmes de refus d'accès liés à la sécurité SELinux.

## RÉSULTATS

Vous allez gagner un peu d'expérience avec les outils de résolution de problèmes de SELinux.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab selinux-issues start**. Cette commande exécute un script de démarrage qui détermine si la machine servera est accessible sur le réseau. Elle installe le service httpd, configure le pare-feu sur servera afin d'autoriser les connexions HTTP et supprime le contexte SELinux pour le répertoire **/custom**.

```
[student@workstation ~]$ lab selinux-issues start
```

- ▶ 1. Utilisez la commande **ssh** pour vous connecter à servera en tant qu'utilisateur student. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur root. Le mot de passe de l'utilisateur student est student.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Ouvrez un navigateur Web sur workstation et tentez d'accéder à **http://servera/index.html**. Vous obtenez un message d'erreur indiquant que vous n'êtes pas autorisé à accéder au fichier.
- ▶ 4. En utilisant la commande **less**, affichez le contenu de **/var/log/messages**. Utilisez la clé / et recherchez **sealert**. Copier le suggéré **sealert** commande de sorte qu'il puisse être utilisé à l'étape suivante. Utilisez **q** pour quitter la commande **less**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 28 06:07:03 servera setroubleshoot[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html. For complete SELinux
messages run: sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
Mar 28 06:07:03 servera platform-python[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html.#012#012***** Plugin
catchall (100. confidence) suggests *****#012#012If
you believe that httpd should be allowed getattr access on the index.html file
by default.#012Then you should report this as a bug.#012You can generate a
local policy module to allow this access.#012Do#012allow this access for now by
executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
-X 300 -i my-httpd.pp#012
Mar 28 06:07:04 servera setroubleshoot[15326]: failed to retrieve rpm info for /
custom/index.html
...output omitted...
```

- ▶ 5. Exécutez la commande **sealert** suggérée. Remarquez le contexte source, les objets cibles, la politique et le mode d'exécution.

```
[root@servera ~]# sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
SELinux is preventing /usr/sbin/httpd from getattr access on the file /custom/
index.html.

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed getattr access on the index.html file
by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:default_t:s0
Target Objects          /custom/index.html [ file ]
Source                 httpd
Source Path             /usr/sbin/httpd
Port                  <Unknown>
Host                  servera.lab.example.com
Source RPM Packages
Target RPM Packages
Policy RPM              selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled         True
Policy Type             targeted
Enforcing Mode          Enforcing
Host Name               servera.lab.example.com
Platform                Linux servera.lab.example.com 4.18.0-67.el8.x86_64
#1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64 x86_64
```

	Alert Count	18
	First Seen	2019-03-25 19:25:28
CET	Last Seen	2019-03-28 11:07:00
CET	Local ID	b1c9cc8f-a953-4625-
b79b-82c4f4f1fee3		
Raw Audit Messages		
<pre>type=AVC msg=audit(1553767620.970:16958):   avc: denied { getattr } for pid=15067 comm="httpd" path="/custom/ index.html" dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0   tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0</pre>		
Hash: httpd,httpd_t,default_t,file,getattr		

- ▶ 6. La section **Raw Audit Messages** de la commande **sealert** contient des informations de **/var/log/audit/audit.log**. Utilisez la commande **ausearch** pour rechercher le fichier **/var/log/audit/audit.log**. L'option **-m** effectue une recherche sur le type de message. L'option **-ts** effectue une recherche fondée sur le temps. Cette entrée identifie le processus et le fichier pertinents à l'origine de l'alerte. Le processus est le serveur Web Apache **httpd**, le fichier est **/custom/index.html** et le contexte est **system\_r:httpd\_t**.

```
[root@servera ~]# ausearch -m AVC -ts recent
-----
time->Thu Mar 28 13:39:30 2019
type=PROCTITLE msg=audit(1553776770.651:17000):
  proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553776770.651:17000): arch=c000003e syscall=257
  success=no exit=-13 a0=fffffff9c a1=7f8db803f598 a2=80000 a3=0 items=0 ppid=15063
  pid=15065 auid=4294967295 uid=48 gid=48 euid=48 suid=48 egid=48
  sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
  subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1553776770.651:17000): avc: denied
  { open } for pid=15065 comm="httpd" path="/custom/index.html"
  dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

- ▶ 7. Pour résoudre le problème, utilisez les commandes **semanage** et **restorecon**. Le contexte à gérer est **httpd\_sys\_content\_t**.

```
[root@servera ~]# semanage fcontext -a -t httpd_sys_content_t '/custom(/.*)?'
[root@servera ~]# restorecon -Rv /custom
Relabeled /custom from unconfined_u:object_r:default_t:s0 to
  unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to
  unconfined_u:object_r:httpd_sys_content_t:s0
```

- ▶ 8. Tentez d'accéder à nouveau <http://servera/index.html>. Vous devez recevoir le message **This is SERVERA**.

▶ 9. Quittez servera.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exécutez le script **lab selinux-issues finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab selinux-issues finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# GESTION DE LA SÉCURITÉ AVEC SELINUX

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans le cadre de cet atelier, vous allez résoudre un problème de refus d'accès par SELinux. Les administrateurs système ne parviennent pas à faire en sorte qu'un nouveau serveur Web fournit du contenu aux clients quand SELinux fonctionne en mode enforcing.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Identifier les problèmes dans les fichiers journaux du système.
- Ajuster la configuration de SELinux.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab selinux-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Il installe également le serveur Apache `httpd`, crée un `DocumentRoot` pour Apache et met à jour le fichier de configuration.

```
[student@workstation ~]$ lab selinux-review start
```

- Connectez-vous à `serverb` en tant qu'utilisateur root.
- Lancez un navigateur Web sur `workstation` et accédez à `http://serverb/lab.html`. Ce message d'erreur s'affiche : **You do not have permission to access / lab.html on this server.**
- Recherchez et identifiez le problème SELinux qui empêche Apache de fournir du contenu Web.
- Affichez le contexte SELinux de la nouvelle racine de document HTTP et de la racine de document HTTP d'origine. Corrigez le problème SELinux qui empêche Apache de fournir du contenu Web.
- Vérifiez que le problème lié à SELinux a été corrigé et qu'Apache peut fournir du contenu Web.
- Quittez `serverb`.

## Évaluation

À partir de `workstation`, exécutez le script `lab selinux-review grade` pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab selinux-review grade
```

## Fin

Sur workstation, exéutez le script **lab selinux-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab selinux-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# GESTION DE LA SÉCURITÉ AVEC SELINUX

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans le cadre de cet atelier, vous allez résoudre un problème de refus d'accès par SELinux. Les administrateurs système ne parviennent pas à faire en sorte qu'un nouveau serveur Web fournit du contenu aux clients quand SELinux fonctionne en mode enforcing.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Identifier les problèmes dans les fichiers journaux du système.
- Ajuster la configuration de SELinux.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab selinux-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Il installe également le serveur Apache `httpd`, crée un `DocumentRoot` pour Apache et met à jour le fichier de configuration.

```
[student@workstation ~]$ lab selinux-review start
```

- Connectez-vous à `serverb` en tant qu'utilisateur root.

- Utilisez la commande `ssh` pour vous connecter à `serverb` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- Utilisez la commande `sudo -i` pour basculer vers l'utilisateur root. Le mot de passe de l'utilisateur `student` est `student`.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

2. Lancez un navigateur Web sur workstation et accédez à `http://serverb/lab.html`. Ce message d'erreur s'affiche : **You do not have permission to access / lab.html on this server.**
3. Recherchez et identifiez le problème SELinux qui empêche Apache de fournir du contenu Web.
  - 3.1. En utilisant la commande `less`, affichez le contenu de `/var/log/messages`. Utilisez la clé `/` et recherchez `sealert`. Utilisez `q` pour quitter la commande `less`.

```
[root@serverb ~]# less /var/log/messages
Mar 28 10:19:51 serverb setroubleshoot[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html. For complete SELinux
messages run: sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
Mar 28 10:19:51 serverb platform-python[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html.#012#012***** Plugin
catchall (100. confidence) suggests *****#012#012If
you believe that httpd should be allowed getattr access on the lab.html file
by default.#012Then you should report this as a bug.#012You can generate a
local policy module to allow this access.#012Do#012allow this access for now by
executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
-X 300 -i my-httpd.pp#012
```

- 3.2. Exécutez la commande `sealert` suggérée. Remarquez le contexte source, les objets cibles, la politique et le mode d'exécution.

```
[root@serverb ~]# sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
SELinux is preventing /usr/sbin/httpd from getattr access on the file /lab-
content/lab.html.

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed getattr access on the lab.html file by
default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp

Additional Information:
Source Context           system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:default_t:s0
Target Objects          /lab-content/lab.html [ file ]
Source                 httpd
Source Path             /usr/sbin/httpd
Port                  <Unknown>
Host                  serverb.lab.example.com
Source RPM Packages
Target RPM Packages    selinux-policy-3.14.1-59.el8.noarch
Policy RPM             selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled         True
Policy Type            targeted
```

```

Enforcing Mode          Enforcing
Host Name               serverb.lab.example.com
Platform                Linux serverb.lab.example.com 4.18.0-67.el8.x86_64
                        #1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64 x86_64
Alert Count              2
First Seen               2019-03-28 15:19:46
CET                      Last Seen           2019-03-28 15:19:46
CET                      Local ID            8824e73d-3ab0-4caf-8258-86e8792fee2d

Raw Audit Messages
type=AVC msg=audit(1553782786.213:864):
avc: denied { getattr } for pid=15606 comm="httpd" path="/lab-content/
lab.html" dev="vda1" ino=8763212 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0

Hash: httpd,httpd_t,default_t,file,getattr

```

- 3.3. La section **Raw Audit Messages** de la commande **sealert** contient des informations de **/var/log/audit/audit.log**. Utilisez la commande **ausearch** pour rechercher le fichier **/var/log/audit/audit.log**. L'option **-m** effectue une recherche sur le type de message. L'option **ts** effectue une recherche fondée sur le temps. Cette entrée identifie le processus et le fichier pertinents à l'origine de l'alerte. Le processus est le serveur Web Apache **httpd**, le fichier est **/lab-content/lab.html** et le contexte est **system\_r:httpd\_t**.

```

[root@serverb ~]# ausearch -m AVC -ts recent
time->Thu Mar 28 15:19:46 2019
type=PROCTITLE msg=audit(1553782786.213:864):
    proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553782786.213:864): arch=c000003e syscall=6 success=no
    exit=-13 a0=7fb900004930 a1=7fb92dfca8e0 a2=7fb92dfca8e0 a3=1 items=0 ppid=15491
    pid=15606 auid=4294967295 uid=48 gid=48 euid=48 suid=48 egid=48
    sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
    subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1553782786.213:864): avc: denied { getattr } for
    pid=15606 comm="httpd" path="/lab-content/lab.html" dev="vda1" ino=8763212
    scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
    tclass=file permissive=0

```

4. Affichez le contexte SELinux de la nouvelle racine de document HTTP et de la racine de document HTTP d'origine. Correz le problème SELinux qui empêche Apache de fournir du contenu Web.
- 4.1. Utilisez la commande **ls -dz** pour comparer la racine de document de **/lab-content** et **/var/www/html**.

```

[root@serverb ~]# ls -dz /lab-content /var/www/html
unconfined_u:object_r:default_t:s0 /lab-content/
system_u:object_r:httpd_sys_content_t:s0 /var/www/html/

```

- 4.2. Créez une règle de contexte de fichier définissant **httpd\_sys\_content\_t** comme type par défaut pour **/lab-content** et pour tous les fichiers situés en aval.

```
[root@serverb ~]# semanage fcontext -a -t httpd_sys_content_t '/lab-content(/.*)?'
```

- 4.3. Utilisez la commande **restorecon** pour définir le contexte SELinux des fichiers de **/lab-content**.

```
[root@serverb ~]# restorecon -R /lab-content/
```

5. Vérifiez que le problème lié à SELinux a été corrigé et qu'Apache peut fournir du contenu Web.

À l'aide de votre navigateur Web, actualisez le lien <http://serverb/lab.html>. Vous devez maintenant voir s'afficher du contenu Web.

```
This is the html file for the SELinux final lab on SERVERB.
```

6. Quittez serverb.

```
[root@serverb ~]# exit  
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

## Évaluation

À partir de **workstation**, exécutez le script **lab selinux-review grade** pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab selinux-review grade
```

## Fin

Sur **workstation**, exécutez le script **lab selinux-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab selinux-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Les commandes **getenforce** et **setenforce** sont utilisées pour gérer le mode SELinux d'un système.
- La commande **semanage** permet de gérer les règles de la politique SELinux. La commande **restorecon** applique le contexte défini par la politique.
- Les valeurs booléennes sont des commutateurs qui modifient le comportement de la politique SELinux. Elles peuvent être activées ou désactivées et sont utilisées pour ajuster la politique.
- La commande **sealert** affiche des informations utiles pour permettre la résolution de problèmes liés à SELinux.

## CHAPITRE 6

# GESTION DU STOCKAGE DE BASE

### PROJET

Créer et gérer des périphériques de stockage, des partitions, des systèmes de fichiers et des espaces d'échange à partir de la ligne de commande.

### OBJECTIFS

- Créer des partitions de stockage, les formater avec des systèmes de fichiers et les monter pour les utiliser.
- Créer et gérer des espaces d'échange pour compléter la mémoire physique.

### SECTIONS

- Ajout de partitions, de systèmes de fichiers et de montages persistants (et exercice guidé)
- Gestion d'espaces de stockage (et exercice guidé)

### ATELIER

Gestion du stockage de base

# AJOUT DE PARTITIONS, DE SYSTÈMES DE FICHIERS ET DE MONTAGES PERSISTANTS

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir créer des partitions de stockage, les formater avec des systèmes de fichiers et les monter pour les utiliser.

## PARTITIONNEMENT D'UN DISQUE

Le partitionnement de disque permet aux administrateurs système de diviser un disque dur en plusieurs unités de stockage logiques, appelées partitions. En divisant un disque en partitions, les administrateurs système peuvent utiliser les partitions pour différentes fonctions.

Par exemple, le partitionnement de disque peut s'avérer nécessaire ou utile dans les situations suivantes :

- limitation de l'espace disponible pour les applications ou pour les utilisateurs ;
- séparation des fichiers de programme et de système d'exploitation des fichiers utilisateur ;
- création d'une zone distincte pour la permutation de mémoire virtuelle ;
- limitation de l'utilisation de l'espace disque pour améliorer les performances des outils de diagnostic et de la création d'images de sauvegarde.

## Schéma de partitionnement MBR

Depuis 1982, le schéma de partitionnement *Master Boot Record (MBR)* régit la manière dont les disques sont partitionnés sur les systèmes basés sur le microprogramme BIOS. Ce schéma prend en charge un maximum de quatre partitions principales. Sur les systèmes Linux, en recourant aux partitions étendues et logiques, les administrateurs peuvent créer jusqu'à 15 partitions. Les données sur la taille des partitions étant stockées sous la forme de valeurs 32 bits, la taille des partitions et des disques est limitée à 2 Tio sur les disques partitionnés selon le schéma MBR.

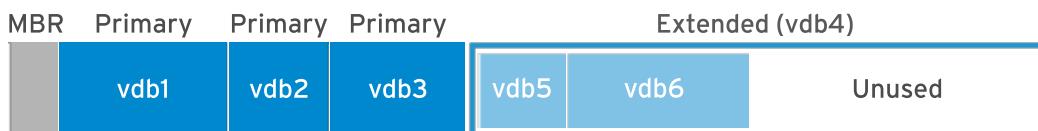


Figure 6.1: Partitionnement MBR du périphérique de stockage /dev/vdb

Dans la mesure où la capacité des disques physiques augmente, et celle des volumes SAN augmentent encore davantage, la limite de 2 Tio de la taille des partitions et des disques du schéma de partitionnement MBR n'est plus une limite théorique, mais plutôt un problème concret que les administrateurs système rencontrent de plus en plus fréquemment dans les environnements de production. En conséquence, le schéma MBR existant est sur le point d'être supplplanté par le nouveau schéma *GUID Partition Table (GPT)* pour le partitionnement des disques.

## Schéma de partitionnement GPT

Sur les systèmes basés sur le microprogramme *Unified Extensible Firmware Interface (UEFI)*, GPT est la norme qui régit la création de tables de partitionnement sur les disques durs physiques. GPT fait partie intégrante de la norme UEFI et surmonte un grand nombre de limites imposées par l'ancien schéma de type MBR.

Un schéma GPT fournit jusqu'à 128 partitions. Contrairement au mode MBR, qui utilise 32 bits pour le stockage des adresses de blocs logiques et pour les informations de taille, un schéma GPT alloue 64 bits pour les adresses de blocs logiques. Ceci permet à un schéma GPT de prendre en charge des partitions et des disques pouvant atteindre huit zébiocents (Zio), soit huit milliards de tébiocents.

En plus de résoudre le problème des limitations propres au schéma de partitionnement MBR, un schéma GPT offre d'autres fonctionnalités et avantages. Un schéma GPT utilise un *identifiant global unique (GUID)* permettant d'identifier chaque disque et partition. Contrairement à un schéma MBR, qui présente un point de défaillance unique, un schéma GPT assure la redondance des informations de sa table de partitionnement. La table GPT principale réside au début du disque, tandis qu'une copie de sauvegarde, la table GPT secondaire, se trouve à la fin du disque. En outre, un schéma GPT recourt aux sommes de contrôle pour détecter les erreurs et les données endommagées dans l'en-tête et la table de partitionnement GPT.



Figure 6.2: Partitionnement GPT du périphérique de stockage /dev/vdb

## GESTION DES PARTITIONS AVEC L'ÉDITEUR PARTED

Les éditeurs de partitions sont des programmes qui permettent aux administrateurs d'apporter des modifications aux partitions d'un disque (ex. : créer ou supprimer des partitions, modifier leur type). Pour effectuer ces opérations, les administrateurs peuvent utiliser l'éditeur de partition parted pour les schémas de partitionnement MBR et GPT.

La commande **parted** prend le nom de périphérique de l'ensemble du disque en tant que premier argument et une ou plusieurs sous-commandes. L'exemple suivant utilise la sous-commande **print** pour afficher la table de partition sur le disque **/dev/vda**.

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  10.7GB  10.7GB  primary   xfs          boot
 2      10.7GB   53.7GB  42.9GB  primary   xfs
```

Si vous ne fournissez pas de sous-commande, **parted** ouvre une session interactive pour émettre des commandes.

## CHAPITRE 6 | Gestion du stockage de base

```
[root@host ~]# parted /dev/vda
GNU Parted 3.2
Using /dev/vda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1       1049kB  10.7GB  10.7GB  primary   xfs          boot
 2       10.7GB  53.7GB  42.9GB  primary   xfs

(parted) quit
[root@host ~]#
```

Par défaut, **parted** affiche toutes les tailles en puissances de 10 (Ko, Mo, Go). Vous pouvez modifier cette configuration par défaut avec la sous-commande **unit** qui accepte les paramètres suivants :

- **s** pour le secteur
- **B** pour l'octet
- **MiB** (Mio), **GiB** (Gio) ou **TiB** (Tio) (puissances de 2)
- **MB** (Mo), **GB** (Go) ou **TB** (To) (puissances de 10)

```
[root@host ~]# parted /dev/vda unit s print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1       2048s   20971486s  20969439s  primary   xfs          boot
 2       20971520s 104857535s  83886016s  primary   xfs
```

Comme indiqué dans l'exemple ci-dessus, vous pouvez également spécifier plusieurs sous-commandes (ici, **unit** et **print**) sur la même ligne.

## Écriture de la table de partition sur un nouveau disque

Pour partitionner un nouveau lecteur, vous devez d'abord lui écrire une étiquette de disque. L'étiquette de disque indique le schéma de partitionnement à utiliser.



### NOTE

Gardez en tête que **parted** applique les modifications immédiatement. Une erreur avec la commande **parted** pourrait certainement conduire à une perte de données.

## CHAPITRE 6 | Gestion du stockage de base

En tant qu'utilisateur **root**, utilisez la commande suivante pour écrire une étiquette de disque MBR sur un disque.

```
[root@host ~]# parted /dev/vdb mklabel msdos
```

Pour écrire une étiquette de disque GPT, utilisez la commande suivante.

```
[root@host ~]# parted /dev/vdb mklabel gpt
```



### MISE EN GARDE

La sous-commande **mklabel** nettoie la table de partition existante. Utilisez uniquement **mklabel** lorsque votre objectif est de réutiliser le disque sans tenir compte des données existantes. Si une nouvelle étiquette modifie les limites de la partition, toutes les données des systèmes de fichiers existants deviennent inaccessibles.

## Création de partitions MBR

La création d'une partition de disque MBR implique plusieurs étapes :

1. Spécifiez le disque sur lequel créer la partition.

En tant qu'utilisateur **root**, exécutez la commande **parted** et spécifiez le nom du disque en argument. Cela lance la commande **parted** en mode interactif et affiche une invite de commande.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. Utilisez la sous-commande **mkpart** pour créer une partition principale ou étendue.

```
(parted) mkpart
Partition type? primary/extended? primary
```



### NOTE

Si plus de quatre partitions sont requises sur un disque partitionné selon le schéma MBR, créez trois partitions principales et une partition étendue. Cette partition étendue agit comme un conteneur dans lequel vous pouvez créer plusieurs partitions logiques.

3. Indiquez le type de système de fichiers que vous voulez créer sur la partition, tel que **xfs** ou **ext4**. Cela ne crée pas le système de fichiers sur la partition ; il s'agit seulement d'une indication du type de partition.

```
File system type? [ext2]? xfs
```

Pour obtenir la liste des types de systèmes de fichiers pris en charge, utilisez la commande suivante :

```
[root@host ~]# parted /dev/vdb help mkpart
mkpart PART-TYPE [FS-TYPE] START END      make a partition

PART-TYPE is one of: primary, logical, extended
FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
hp-ufs, sun-ufs, xfs, apfs2, apfs1, afs, amufs5, amufs4, amufs3,
amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
START and END are disk locations, such as 4GB or 10%. Negative values
count from the end of the disk. For example, -1s specifies exactly the
last sector.

'mkpart' makes a partition without creating a new file system on the
partition. FS-TYPE may be specified to set an appropriate partition
ID.
```

- Spécifiez le secteur du disque au niveau duquel la nouvelle partition doit démarrer.

Start? **2048s**

Remarquez le suffixe **s** qui fournit la valeur dans les secteurs. Vous pouvez également utiliser les suffixes **MiB**, **GiB**, **TiB**, **MB**, **GB** ou **TB**. Si vous ne fournissez pas de suffixe, **MB** est la valeur par défaut. **parted** peut arrondir la valeur que vous fournissez pour satisfaire les contraintes du disque.

Lorsque **parted** démarre, il récupère la topologie du disque du périphérique. Par exemple, la taille de bloc physique du disque est généralement un paramètre qui est collecté par **parted**. Avec cette information, **parted** garantit que la position de départ que vous indiquez aligne correctement la partition sur la structure du disque. Un alignement correct des partitions est important pour des performances optimales. Si la position de départ entraîne un problème d'alignement de la partition, **parted** affiche un avertissement. Avec la plupart des disques, un secteur de départ, multiple de 2048, constitue une hypothèse sûre.

- Spécifiez le secteur du disque où la nouvelle partition doit se terminer.

End? **1000MB**

Avec **parted**, vous ne pouvez pas fournir directement la taille de votre partition, mais vous pouvez la calculer rapidement avec la formule suivante :

**Size = End - Start**

Dès que vous fournissez la position de fin, **parted** met à jour la table de partition sur le disque avec les détails de la nouvelle partition.

- Quittez **parted**.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

- Exécutez la commande **udevadm settle**. Cette commande attend que le système détecte la nouvelle partition et crée le fichier de périphérique associé sous le répertoire **/dev**. Elle ne revient que lorsque l'opération est terminée.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

À la place du mode interactif, vous pouvez également créer la partition comme suit :

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

## Création de partitions GPT

Le schéma GPT utilise également la commande **parted** pour créer des partitions :

- Spécifiez le disque sur lequel créer la partition.

En tant qu'utilisateur **root**, exécutez la commande **parted** avec le disque comme seul argument pour démarrer **parted** en mode interactif avec une invite de commande.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

- Utilisez la sous-commande **mkpart** pour commencer à créer la partition.

Avec le schéma GPT, chaque partition se voit attribuer un nom.

```
(parted) mkpart
Partition name? []? usersdata
```

- Indiquez le type de système de fichiers que vous voulez créer sur la partition, tel que **xfs** ou **ext4**. Cela ne crée pas le système de fichiers sur la partition ; il s'agit seulement d'une indication du type de partition.

```
File system type? [ext2]? xfs
```

- Spécifiez le secteur du disque au niveau duquel la nouvelle partition doit démarrer.

```
Start? 2048s
```

- Spécifiez le secteur du disque où la nouvelle partition doit se terminer.

```
End? 1000MB
```

Dès que vous fournissez la position de fin, **parted** met à jour la table de partition sur le disque avec les détails de la nouvelle partition.

#### 6. Quittez **parted**.

```
(parted) quit  
Information: You may need to update /etc/fstab.  
[root@host ~]#
```

#### 7. Exécutez la commande **udevadm settle**. Cette commande attend que le système détecte la nouvelle partition et crée le fichier de périphérique associé sous le répertoire **/dev**. Elle ne revient que lorsque l'opération est terminée.

```
[root@host ~]# udevadm settle  
[root@host ~]#
```

À la place du mode interactif, vous pouvez également créer la partition comme suit :

```
[root@host ~]# parted /dev/vdb mkpart usersdata xfs 2048s 1000MB
```

## Suppression de partitions

Les étapes suivantes s'appliquent aux schémas de partitionnement MBR et GPT.

#### 1. Spécifiez le disque qui contient la partition à supprimer.

En tant qu'utilisateur **root**, exécutez la commande **parted** avec le disque comme seul argument pour démarrer **parted** en mode interactif avec une invite de commande.

```
[root@host ~]# parted /dev/vdb  
GNU Parted 3.2  
Using /dev/vdb  
Welcome to GNU Parted! Type 'help' to view a list of commands.  
(parted)
```

#### 2. Identifiez le numéro de la partition à supprimer.

```
(parted) print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:  
  
Number  Start   End     Size   File system  Name      Flags  
 1       1049kB  1000MB  999MB  xfs          usersdata
```

#### 3. Supprimez la partition.

```
(parted) rm 1
```

La sous-commande **rm** supprime immédiatement la partition de la table de partitions sur le disque.

#### 4. Quittez **parted**.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

## CRÉATION DE SYSTÈMES DE FICHIERS

Après la création d'un périphérique en mode bloc, l'étape suivante consiste à lui ajouter un système de fichiers. Red Hat Enterprise Linux prend en charge de nombreux types de systèmes de fichiers, mais les deux types les plus courants sont XFS and ext4. Anaconda, le programme d'installation de Red Hat Enterprise Linux, utilise XFS par défaut.

En tant qu'utilisateur **root**, utilisez la commande **mkfs.xfs** pour appliquer un système de fichiers à un périphérique en mode bloc. Pour ext4, utilisez **mkfs.ext4**.

```
[root@host ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=60992 blks
                                =          sectsz=512  attr=2, projid32bit=1
                                =          crc=1      finobt=1, sparse=1, rmapbt=0
                                =          reflink=1
data     =          bsize=4096   blocks=243968, imaxpct=25
        =          sunit=0     swidth=0 blks
naming  =version 2              bsize=4096   ascii-ci=0, ftype=1
log     =internal log           bsize=4096   blocks=1566, version=2
        =          sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
```

## MONTAGE DE SYSTÈMES DE FICHIERS

Une fois que vous avez ajouté le système de fichiers, la dernière étape consiste à monter le système de fichiers dans un répertoire de la structure de répertoires. Lorsque vous montez un système de fichiers sur l'arborescence des répertoires, les utilitaires d'espace utilisateur peuvent lire ou écrire des fichiers sur le périphérique.

### Montage manuel de systèmes de fichiers

Les administrateurs utilisent la commande **mount** pour connecter manuellement le périphérique à un emplacement de répertoire ou point de montage. Il faut indiquer dans la commande **mount** le périphérique, le point de montage et éventuellement les options du système de fichiers comme arguments. Les options du système de fichiers personnalisent le comportement du système de fichiers.

```
[root@host ~]# mount /dev/vdb1 /mnt
```

Vous pouvez également utiliser la commande **mount** pour visualiser les systèmes de fichiers actuellement montés, leurs points de montage et leurs options.

```
[root@host ~]# mount | grep vdb1  
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

## Montage persistant de systèmes de fichiers

Le montage manuel d'un système de fichiers constitue un excellent moyen de s'assurer qu'un périphérique formaté est accessible ou qu'il fonctionne comme il faut. Toutefois, lorsque le serveur redémarre, le système ne monte pas automatiquement le système de fichiers sur l'arborescence de répertoires. Les données sont intactes sur le système de fichiers, mais les utilisateurs ne peuvent pas y accéder.

Pour vous assurer que le système monte automatiquement le système de fichiers au démarrage du système, ajoutez une entrée au fichier **/etc/fstab**. Ce fichier de configuration répertorie les systèmes de fichiers à monter au démarrage du système.

**/etc/fstab** est un fichier délimité par des espaces qui contient six champs par ligne.

```
[root@host ~]# cat /etc/fstab  
  
#  
# /etc/fstab  
# Created by anaconda on Wed Feb 13 16:39:59 2019  
#  
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.  
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.  
#  
# After editing this file, run 'systemctl daemon-reload' to update systemd  
# units generated from this file.  
#  
UUID=a8063676-44dd-409a-b584-68be2c9f5570    /          xfs    defaults    0 0  
UUID=7a20315d-ed8b-4e75-a5b6-24ff9e1f9838    /dbdata    xfs    defaults    0 0
```

Lorsque vous ajoutez ou supprimez une entrée dans le fichier **/etc/fstab**, exécutez la commande **systemctl daemon-reload** ou redémarrez le serveur, afin que **systemd** enregistre la nouvelle configuration.

```
[root@host ~]# systemctl daemon-reload
```

Le *premier champ* spécifie le périphérique. Cet exemple utilise l'UUID pour spécifier le périphérique. Les systèmes de fichiers créent et stockent l'UUID dans leur superblock au moment de la création. Vous pouvez également utiliser le fichier de périphérique, tel que **/dev/vdb1**.

**NOTE**

Il est préférable d'utiliser l'UUID, car les identifiants d'un périphérique en mode bloc peuvent changer dans certains cas, par exemple lorsqu'un fournisseur de services de cloud modifie la couche de stockage sous-jacente d'une machine virtuelle, ou si les disques sont détectés dans un ordre différent à chaque démarrage de système. Le nom de fichier du périphérique en mode bloc peut changer, mais l'UUID reste constant dans le superblock du système de fichiers.

Utilisez la commande **lsblk --fs** pour analyser les périphériques en mode bloc connectés à une machine et récupérer les UUID du système de fichiers.

```
[root@host ~]# lsblk --fs
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
sr0
vda
└─vda1  xfs   a8063676-44dd-409a-b584-68be2c9f5570 /
vdb
└─vdb1  xfs   7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata
```

Le deuxième champ est le point de montage du répertoire à partir duquel le périphérique en mode bloc est accessible dans la structure de répertoires. Le point de montage doit exister ; sinon, créez-le avec la commande **mkdir**.

Le troisième champ contient le type de système de fichiers, tel que **xfs** ou **ext4**.

Le quatrième champ est la liste des options séparées par des virgules à appliquer au périphérique. **defaults** est un ensemble d'options couramment utilisées. La page de manuel **mount(8)** documente les autres options disponibles.

Le cinquième champ est utilisé par la commande **dump** pour sauvegarder le périphérique. Les autres applications de sauvegarde n'utilisent généralement pas ce champ.

Le dernier champ, le champ d'ordre **fsck**, détermine si la commande **fsck** doit être exécutée au démarrage du système pour vérifier que les systèmes de fichiers sont propres. La valeur dans ce champ indique l'ordre dans lequel **fsck** doit être exécutée. Pour les systèmes de fichiers XFS, définissez ce champ sur **0**, car XFS n'utilise pas **fsck** pour vérifier le statut du système de fichiers. Dans le cas des systèmes de fichiers ext4, définissez-le sur **1** pour le système de fichiers root et sur **2** pour les autres systèmes de fichiers ext4. De cette façon, **fsck** traite d'abord le système de fichiers root, puis vérifie les systèmes de fichiers sur des disques distincts simultanément, et les systèmes de fichiers sur le même disque en séquence.

**NOTE**

La présence d'une entrée incorrecte dans le fichier **/etc/fstab** est susceptible d'empêcher le démarrage de la machine. Les administrateurs doivent vérifier la validité de l'entrée en démontant le nouveau système de fichiers, puis en lançant **mount /mountpoint** qui lit le fichier **/etc/fstab**, afin de remonter le système de fichiers. Si la commande **mount** renvoie un message d'erreur, il convient de corriger le problème avant de redémarrer la machine.

Vous pouvez également utiliser la commande **findmnt --verify** pour contrôler le fichier **/etc/fstab**.



## RÉFÉRENCES

**info parted**(*Manuel utilisateur GNU Parted*)

Pages du manuel **parted**(8), **mkfs**(8), **mount**(8), **lsblk**(8) et **fstab**(5)

Pour plus d'informations, consultez le guide *Configuring and managing file systems* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_and\\_managing\\_file\\_systems/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_file_systems/)

## ► EXERCICE GUIDÉ

# AJOUT DE PARTITIONS, DE SYSTÈMES DE FICHIERS ET DE MONTAGES PERSISTANTS

Dans cet exercice, vous allez créer une partition sur un nouveau périphérique de stockage, la formater avec un système de fichiers XFS, la configurer pour qu'elle soit montée au démarrage et la monter pour l'utiliser.

## RÉSULTATS

Vous devez pouvoir utiliser **parted**, **mkfs.xfs** et d'autres commandes permettant de créer une partition sur un nouveau disque, de la formater et de la monter de manière persistance.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab storage-partitions start**. Cette commande exécute un script de démarrage qui détermine si la machine servera est accessible sur le réseau. Elle prépare également le second disque sur servera pour l'exercice.

```
[student@workstation ~]$ lab storage-partitions start
```

- ▶ 1. Utilisez la commande **ssh** pour vous connecter à servera en tant qu'utilisateur student. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur root. Si vous y êtes invité, utilisez le mot de passe student.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Utilisez la commande **parted** pour créer une étiquette de disque de type **msdos** sur le disque **/dev/vdb** pour préparer ce dernier pour le schéma de partitionnement MBR.

**CHAPITRE 6 |** Gestion du stockage de base

```
[root@servera ~]# parted /dev/vdb mklabel msdos
Information: You may need to update /etc/fstab.
```

- 4. Ajoutez une partition principale d'1 Go. Pour un alignement correct, démarrez la partition au secteur 2048. Définissez le type de système de fichiers de la partition sur XFS.

4.1. Utilisez **parted** en mode interactif pour vous aider à créer la partition.

```
[root@servera ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? xfs
Start? 2048s
End? 1001MB
(parted) quit
Information: You may need to update /etc/fstab.
```

Étant donné que la partition commence au secteur 2048, la commande précédente définit la position de fin sur 1001 Mo pour obtenir une taille de partition de 1000 Mo (1 Go).

Vous pouvez également effectuer la même opération avec la commande non interactive suivante : **parted /dev/vdb mkpart primary xfs 2048s 1001MB**

4.2. Vérifiez votre travail en listant les partitions sur **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1       1049kB  1001MB  1000MB  primary
```

4.3. Exécutez la commande **udevadm settle**. Cette commande attend que le système enregistre la nouvelle partition et revient lorsque l'opération est terminée.

```
[root@servera ~]# udevadm settle
[root@servera ~]#
```

- 5. Formatez la nouvelle partition avec le système de fichiers XFS.

```
[root@servera ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=61056 blks
                                =                      sectsz=512  attr=2, projid32bit=1
                                =                      crc=1      finobt=1, sparse=1, rmapbt=0
                                =                      reflink=1
```

**CHAPITRE 6 |** Gestion du stockage de base

```
data      =          bsize=4096  blocks=244224, imaxpct=25
          =
naming   =version 2    sunit=0    swidth=0 blks
log     =internal log  bsize=4096  ascii-ci=0, ftype=1
          =
realtime =none        sectsz=512  sunit=0 blks, lazy-count=1
                      extsz=4096  blocks=0, rtextents=0
```

- 6. Configurez le nouveau système de fichiers pour qu'il soit monté de façon persistante dans **/archive**.

- 6.1. Utilisez **mkdir** pour créer le répertoire **/archive**, qui sera utilisé comme point de montage.

```
[root@servera ~]# mkdir /archive
[root@servera ~]#
```

- 6.2. Utilisez la commande **lsblk** avec l'option **--fs** pour détecter l'UUID du périphérique **/dev/vdb1**.

```
[root@servera ~]# lsblk --fs /dev/vdb
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
vdb
└─vdb1  xfs   e3db1abe-6d96-4faa-a213-b96a6f85dcc1
```

L'UUID de la sortie précédente est probablement différent sur votre système.

- 6.3. Ajoutez une entrée à **/etc/fstab**. Dans la commande suivante, remplacez l'UUID par celui que vous avez détecté à l'étape précédente.

```
[root@servera ~]# vim /etc/fstab
...output omitted...
UUID=e3db1abe-6d96-4faa-a213-b96a6f85dcc1  /archive  xfs  defaults  0  0
```

- 6.4. Mettez à jour **systemd** pour que le système enregistre la nouvelle configuration **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
[root@servera ~]#
```

- 6.5. Exécutez la commande **mount /archive** pour monter le nouveau système de fichiers à l'aide de la nouvelle entrée ajoutée dans **/etc/fstab**.

```
[root@servera ~]# mount /archive
[root@servera ~]#
```

- 6.6. Vérifiez que le nouveau système de fichiers est monté dans **/archive**.

```
[root@servera ~]# mount | grep /archive
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

- 7. Redémarrez **servera**. Après avoir redémarré le serveur, ouvrez une session et vérifiez que **/dev/vdb1** est monté dans **/archive**. Lorsque vous avez terminé, déconnectez-vous de **servera**.

7.1. Redémarrez **servera**.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

7.2. Attendez quelques minutes que **servera** redémarre, puis connectez-vous en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

7.3. Vérifiez que **/dev/vdb1** est monté dans **/archive**.

```
[student@servera ~]$ mount | grep /archive
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

7.4. Déconnectez-vous de **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur **workstation**, exécutez le script **lab storage-partitions finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab storage-partitions finish
```

L'exercice guidé est maintenant terminé.

# GESTION DE L'ESPACE D'ÉCHANGE

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir créer et gérer des espaces d'échange pour compléter la mémoire physique.

## PRÉSENTATION DES CONCEPTS LIÉS À L'ESPACE D'ÉCHANGE

Un espace d'échange est une zone sur un disque qui est contrôlée par le sous-système de gestion de la mémoire du noyau Linux. Le noyau utilise l'espace d'échange pour compléter la RAM du système en faisant transiter les pages de mémoire inactives. La combinaison de la RAM du système et de l'espace d'échange est appelée *mémoire virtuelle*.

Lorsque la mémoire utilisée sur un système dépasse une limite définie, le noyau effectue une recherche dans la mémoire vive pour y trouver les pages de mémoire inactives qui sont affectées à des processus. Le noyau écrit alors les pages inactives dans la zone d'échange et réaffecte les pages RAM afin qu'elles soient utilisées par d'autres processus. Si un programme demande à accéder à une page écrite sur le disque, le noyau localise une autre page de mémoire inactive, l'écrit sur le disque et rappelle la page demandée depuis la zone d'échange.

Comme les zones d'échange sont sur disque, la mémoire virtuelle est plus lente que la RAM. Bien qu'il soit utilisé pour augmenter la RAM système, vous ne devez pas considérer l'espace d'échange comme une solution durable dans le cas où la mémoire RAM est insuffisante pour traiter votre charge de travail.

## Dimensionnement de l'espace d'échange

Les administrateurs doivent dimensionner l'espace d'échange en fonction de la charge de travail de la mémoire sur le système. Les fournisseurs d'applications effectuent parfois des recommandations à ce sujet. Le tableau suivant fournit des indications en fonction de la quantité totale de mémoire physique.

### Recommandations concernant la RAM et l'espace d'échange

DE RAM	ESPACE D'ÉCHANGE	ESPACE D'ÉCHANGE SI L'HIBERNATION EST PERMISE
2 Gio ou moins	Deux fois la RAM	Trois fois la RAM
Entre 2 Gio et 8 Gio	Identique à la RAM	Deux fois la RAM
Entre 8 Gio et 64 Gio	Au moins 4 Gio	1,5 fois la RAM
Plus de 64 Gio	Au moins 4 Gio	L'hibernation n'est pas recommandée

## CHAPITRE 6 | Gestion du stockage de base

La fonction d'hibernation des ordinateurs portables et des ordinateurs de bureau utilise l'espace d'échange pour enregistrer le contenu de la RAM avant de mettre le système hors tension. Lorsque vous rallumez le système, le noyau restaure le contenu de la RAM à partir de l'espace d'échange et ne nécessite pas un démarrage complet. Pour ces systèmes, l'espace d'échange doit être supérieur à la quantité de RAM.

L'article de la base de connaissances de la section Référence à la fin de cette section fournit des indications supplémentaires sur le dimensionnement de l'espace d'échange.

## CRÉATION D'UN ESPACE D'ÉCHANGE

Pour créer un espace d'échange, vous devez procéder comme suit :

- Créez une partition avec le type de système de fichiers **linux-swap**.
- Formatez une signature d'échange sur le périphérique.

### Création d'une partition d'échange

Utilisez **parted** pour créer une partition de la taille souhaitée et définissez son type de système de fichiers sur **linux-swap**. Auparavant, des outils examinaient le type de système de fichiers de la partition pour déterminer si le périphérique devait être activé, mais ce n'est plus le cas. Bien que les utilitaires n'utilisent plus le type de système de fichiers de la partition, sa définition permet aux administrateurs de déterminer rapidement le rôle de la partition.

L'exemple suivant crée une partition de 256 Mo.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB 1001MB 1000MB          data

(parted) mkpart
Partition name? []? swap1
File system type? [ext2]? linux-swap
Start? 1001MB
End? 1257MB
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB 1001MB 1000MB          data
 2      1001MB 1257MB 256MB   linux-swap(v1)  swap1
```

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

Après avoir créé la partition, exécutez la commande **udevadm settle**. Cette commande attend que le système détecte la nouvelle partition et crée le fichier de périphérique associé dans **/dev**. Elle ne revient que lorsque l'opération est terminée.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

## Formatage du périphérique

La commande **mkswap** applique une signature d'échange au périphérique. Contrairement à d'autres utilitaires de formatage, **mkswap** écrit un bloc de données unique au début du périphérique, et laisse le reste du périphérique non formaté pour que le noyau puisse l'utiliser pour le stockage de pages mémoire.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

## ACTIVATION D'UN ESPACE D'ÉCHANGE

Vous pouvez utiliser la commande **swapon** pour activer un espace d'échange formaté.

Utilisez **swapon** avec le périphérique en tant que paramètre ou utilisez **swapon -a** pour activer tous les espaces d'échange listés dans le fichier **/etc/fstab**. Utilisez les commandes **swapon --show** et **free** pour examiner les espaces d'échange disponibles.

```
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:       1873036       134688      1536436          16748      201912       1576044
Swap:          0          0          0
[root@host ~]# swapon /dev/vdb2
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:       1873036       135044      1536040          16748      201952       1575680
Swap:      249852           0      249852
```

Vous pouvez désactiver un espace d'échange à l'aide de la commande **swapoff**. Si des pages sont écrites sur celui-ci, **swapoff** tente de déplacer ces pages vers d'autres espaces d'échange actifs ou de les remettre en mémoire. Si elle ne parvient pas à écrire les données dans d'autres emplacements, la commande **swapoff** échoue avec un message d'erreur et l'espace d'échange reste actif.

## Activation de l'espace d'échange de manière persistante

Pour activer un espace d'échange à chaque démarrage, placez une entrée dans le fichier **/etc/fstab**. L'exemple ci-dessous montre une ligne typique dans **/etc/fstab** basée sur l'espace d'échange créé ci-dessus.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881 swap swap defaults 0 0
```

Le *premier champ* utilisé dans l'exemple ci-dessous est l'UUID. Lorsque vous formatez le périphérique, la commande **mkswap** affiche cet UUID. Si vous avez perdu la sortie de **mkswap**, utilisez la commande **lsblk --fs**. Vous pouvez également utiliser le nom du périphérique dans le premier champ.

Le *deuxième champ* est généralement réservé au point de montage. Toutefois, dans le cas des périphériques d'échange, qui ne sont pas accessibles par le biais de la structure des répertoires, ce champ prend une valeur de remplissage : **swap**.

Le *troisième champ* représente le type du système de fichiers. Le type de système de fichiers d'un espace d'échange est **swap**.

Le *quatrième champ* est réservé aux options. L'exemple utilise l'option **defaults**. L'option **defaults** inclut l'option de montage **auto**. C'est celle-là même qui active automatiquement l'espace d'échange au démarrage.

Les deux derniers champs correspondent à l'indicateur **dump** et à l'ordre **fsck**. Les espaces d'échange ne nécessitent ni sauvegarde ni vérification du système de fichiers, ces deux champs doivent être définis sur zéro.

Lorsque vous ajoutez ou supprimez une entrée dans le fichier **/etc/fstab**, exécutez la commande **systemctl daemon-reload** ou redémarrez le serveur, afin que systemd enregistre la nouvelle configuration.

```
[root@host ~]# systemctl daemon-reload
```

## Définition de la priorité des espaces d'échange

Par défaut, le système utilise des espaces d'échange de manière séquentielle : le noyau utilise le premier espace d'échange activé jusqu'à ce qu'il soit plein, puis il commence à utiliser le second espace d'échange. Cependant, vous pouvez définir une priorité pour chaque espace d'échange afin de forcer cet ordre.

Pour définir la priorité, utilisez l'option **pri** dans **/etc/fstab**. Le noyau utilise d'abord l'espace d'échange dont la priorité est la plus élevée. La priorité par défaut est -2.

L'exemple suivant montre trois espaces d'échange définis dans **/etc/fstab**. Le noyau utilise la dernière entrée en premier, avec **pri=10**. Lorsque cet espace est plein, il utilise la deuxième entrée, avec **pri=4**. Enfin, il utilise la première entrée, dont la priorité par défaut est -2.

```
UUID=af30cbb0-3866-466a-825a-58889a49ef33 swap swap defaults 0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881 swap swap pri=4 0 0
UUID=fb7fa60-b781-44a8-961b-37ac3ef572bf swap swap pri=10 0 0
```

Utilisez **swapon --show** pour afficher les priorités des espaces d'échange.

Lorsque les espaces d'échange ont la même priorité, le noyau les écrit en circuit cyclique.



## RÉFÉRENCES

Pages de manuel **mkswap(8)**, **swapon(8)**, **swapoff(8)**, **mount(8)** et **parted(8)**

**Base de connaissances : quelle est la taille d'échange recommandée pour les plates-formes Red Hat ?**

<https://access.redhat.com/solutions/15244>

## ► EXERCICE GUIDÉ

# GESTION DE L'ESPACE D'ÉCHANGE

Dans cet exercice, vous allez créer et formater une partition à utiliser comme espace d'échange, la formater en tant que swap et l'activer de manière persistante.

## RÉSULTATS

Vous devez pouvoir créer une partition et un espace d'échange sur un disque à l'aide du schéma de partitionnement GPT.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab storage-swap start**. Cette commande exécute un script de démarrage qui détermine si la machine servera est accessible sur le réseau. Elle prépare également le second disque sur servera pour l'exercice.

```
[student@workstation ~]$ lab storage-swap start
```

- ▶ 1. Utilisez la commande **ssh** pour vous connecter à servera en tant qu'utilisateur student. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur root. Si vous y êtes invité, utilisez le mot de passe student.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Utilisez la commande **parted** pour examiner le disque **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	1001MB	1000MB		data	

Notez que le disque a déjà une table de partitions et utilise le schéma de partitionnement GPT. En outre, une partition d'1 Go existe déjà.

- ▶ 4. Ajoutez une nouvelle partition de 500 Mo à utiliser en tant qu'espace d'échange. Définissez le type de partition **linux-swap**.
  - 4.1. Utilisez **parted** pour créer la partition. Comme le disque utilise le schéma de partitionnement GPT, vous devez lui attribuer un nom. Appelez-la **myswap**.

```
[root@servera ~]# parted /dev/vdb mkpart myswap linux-swap 1001MB 1501MB
Information: You may need to update /etc/fstab.
```

Notez dans la commande précédente que la position de départ, 1 001 Mo, correspond à la fin de la première partition existante. De cette façon, **parted** s'assure que la nouvelle partition suit immédiatement la précédente, sans espace.

Étant donné que la partition commence à la position 1 001 Mo, la commande définit la position de fin sur 1 501 Mo pour obtenir une taille de partition de 500 Mo.

- 4.2. Vérifiez votre travail en listant les partitions sur **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  1001MB  1000MB  data
 2      1001MB  1501MB  499MB   myswap  swap
```

La taille de la nouvelle partition n'est pas exactement 500 Mo. Ceci est dû au fait que **parted** doit aligner la partition sur le schéma du disque.

- 4.3. Exécutez la commande **udevadm settle**. Cette commande attend que le système enregistre la nouvelle partition et revient lorsque l'opération est terminée.

```
[root@servera ~]# udevadm settle
[root@servera ~]#
```

- ▶ 5. Initialisez la partition que vous venez de créer en tant qu'espace d'échange.

```
[root@servera ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 476 MiB (499118080 bytes)
no label, UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328
```

- ▶ 6. Activez l'espace d'échange que vous venez de créer.

**CHAPITRE 6 |** Gestion du stockage de base

- 6.1. Utilisez la commande **swapon --show** pour montrer que la création et l'initialisation d'un espace d'échange ne suffisent pas à permettre son utilisation.

```
[root@servera ~]# swapon --show  
[root@servera ~]#
```

- 6.2. Activez l'espace d'échange que vous venez de créer.

```
[root@servera ~]# swapon /dev/vdb2  
[root@servera ~]#
```

- 6.3. Vérifiez que l'espace d'échange que vous venez de créer est désormais disponible.

```
[root@servera ~]# swapon --show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2 partition 476M   0B   -2
```

- 6.4. Désactivez l'espace d'échange.

```
[root@servera ~]# swapoff /dev/vdb2  
[root@servera ~]#
```

- 6.5. Vérifiez que l'espace d'échange est désactivé.

```
[root@servera ~]# swapon --show  
[root@servera ~]#
```

► 7. Configurez le nouvel espace d'échange à activer au démarrage du système.

- 7.1. Utilisez la commande **lsblk** avec l'option **--fs** pour détecter l'UUID du périphérique **/dev/vdb2**.

```
[root@servera ~]# lsblk --fs /dev/vdb2  
NAME FSTYPE LABEL UUID                                     MOUNTPOINT  
vdb2 swap          cb7f71ca-ee82-430e-ad4b-7dda12632328
```

L'UUID de la sortie précédente est probablement différent sur votre système.

- 7.2. Ajoutez une entrée à **/etc/fstab**. Dans la commande suivante, remplacez l'UUID par celui que vous avez détecté à l'étape précédente.

```
[root@servera ~]# vim /etc/fstab  
...output omitted...  
UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328    swap    swap    defaults    0    0
```

- 7.3. Mettez à jour **systemd** pour que le système enregistre la nouvelle configuration **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload  
[root@servera ~]#
```

**CHAPITRE 6 |** Gestion du stockage de base

- 7.4. Activez l'espace d'échange à l'aide de l'entrée que vous venez d'ajouter au fichier **/etc/fstab**.

```
[root@servera ~]# swapon -a  
[root@servera ~]#
```

- 7.5. Vérifiez que le nouvel espace d'échange a été activé.

```
[root@servera ~]# swapon --show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2 partition 476M   0B   -2
```

- 8. Redémarrez **servera**. Une fois le serveur redémarré, ouvrez une session et vérifiez que l'espace d'échange est activé. Lorsque vous avez terminé, déconnectez-vous de **servera**.

- 8.1. Redémarrez **servera**.

```
[root@servera ~]# systemctl reboot  
Connection to servera closed by remote host.  
Connection to servera closed.  
[student@workstation ~]$
```

- 8.2. Attendez quelques minutes que **servera** redémarre, puis connectez-vous en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 8.3. Vérifiez que l'espace d'échange a été activé.

```
[root@servera ~]# swapon --show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2 partition 476M   0B   -2
```

- 8.4. Déconnectez-vous de **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur **workstation**, exécutez le script **lab storage-swap finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab storage-swap finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# GESTION DU STOCKAGE DE BASE

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez créer plusieurs partitions sur un nouveau disque. Vous formaterez certaines de ces partitions avec les systèmes de fichiers et les monterez, puis vous en activerez d'autres en tant qu'espaces d'échange.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Afficher et créer des partitions en utilisant la commande **parted**.
- Créer des systèmes de fichiers sur des partitions et les monter de manière persistante.
- Créer et activer des espaces d'échange au démarrage.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant qu'utilisateur **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab storage-review start**. Cette commande exécute un script de démarrage qui détermine si la machine **serverb** est accessible sur le réseau. Elle prépare également le second disque sur **serverb** pour l'exercice.

```
[student@workstation ~]$ lab storage-review start
```

1. De nouveaux disques sont disponibles sur **serverb**. Sur le premier de ces nouveaux disques, créez une partition GPT de 2 Go nommée **backup**. Comme il peut être difficile de définir la taille exacte, une taille comprise entre 1,8 et 2,2 Go est acceptable. Définissez le type de système de fichiers approprié sur cette partition pour héberger un système de fichiers XFS.  
Le mot de passe du compte d'utilisateur **student** sur **serverb** est **student**. Cet utilisateur dispose d'un accès **root** complet avec la commande **sudo**.
2. Formatez la partition de 2 Go avec un système de fichiers XFS et montez-la de manière persistante dans **/backup**.
3. Sur ce même nouveau disque, créez deux partitions GPT de 512 Mo nommées **swap1** et **swap2**. Une taille entre 460 et 564 Mo est acceptable. Définissez le type de système de fichiers approprié sur ces partitions pour héberger des espaces d'échange.
4. Initialisez les deux partitions de 512 Mo en tant qu'espaces d'échange et configurez-les de sorte qu'elles soient activées au démarrage. Définissez l'espace d'échange **swap2** comme étant prioritaire sur l'autre.
5. Pour vérifier votre travail, redémarrez **serverb**. Vérifiez que le système monte automatiquement la première partition dans **/backup**. Vérifiez également que le système active les deux espaces d'échange.

Lorsque vous avez terminé, déconnectez-vous de serverb.

## Évaluation

À partir de workstation, exécutez le script **lab storage-review grade** pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab storage-review grade
```

## Fin

Sur workstation, exécutez le script **lab storage-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab storage-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# GESTION DU STOCKAGE DE BASE

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez créer plusieurs partitions sur un nouveau disque. Vous formaterez certaines de ces partitions avec les systèmes de fichiers et les monterez, puis vous en activerez d'autres en tant qu'espaces d'échange.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Afficher et créer des partitions en utilisant la commande **parted**.
- Créer des systèmes de fichiers sur des partitions et les monter de manière persistante.
- Créer et activer des espaces d'échange au démarrage.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant qu'utilisateur **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab storage-review start**. Cette commande exécute un script de démarrage qui détermine si la machine **serverb** est accessible sur le réseau. Elle prépare également le second disque sur **serverb** pour l'exercice.

```
[student@workstation ~]$ lab storage-review start
```

1. De nouveaux disques sont disponibles sur **serverb**. Sur le premier de ces nouveaux disques, créez une partition GPT de 2 Go nommée **backup**. Comme il peut être difficile de définir la taille exacte, une taille comprise entre 1,8 et 2,2 Go est acceptable. Définissez le type de système de fichiers approprié sur cette partition pour héberger un système de fichiers XFS. Le mot de passe du compte d'utilisateur **student** sur **serverb** est **student**. Cet utilisateur dispose d'un accès **root** complet avec la commande **sudo**.

- 1.1. Utilisez la commande **ssh** pour vous connecter à **serverb** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Dans la mesure où la création de partitions et de systèmes de fichiers nécessite un accès **root**, utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Si vous y êtes invité, utilisez le mot de passe **student**.

**CHAPITRE 6 |** Gestion du stockage de base

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 1.3. Utilisez la commande **lsblk** pour identifier les nouveaux disques. Ces disques ne doivent comporter aucune partition pour le moment.

```
[root@serverb ~]# lsblk  
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sr0     11:0    1 1024M  0 rom  
vda    252:0    0   10G  0 disk  
└─vda1 252:1    0   10G  0 part /  
vdb   252:16   0    5G  0 disk  
vdc    252:32   0    5G  0 disk  
vdd    252:48   0    5G  0 disk
```

Notez que le premier nouveau disque, **vdb**, n'a aucune partition.

- 1.4. Vérifiez que le disque n'a pas d'étiquette.

```
[root@serverb ~]# parted /dev/vdb print  
Error: /dev/vdb: unrecognised disk label  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B  
Partition Table: unknown  
Disk Flags:
```

- 1.5. Utilisez les sous-commandes **parted** et **mklabel** pour définir le schéma de partitionnement GPT.

```
[root@serverb ~]# parted /dev/vdb mkllabel gpt  
Information: You may need to update /etc/fstab.
```

- 1.6. Créez la partition de 2 Go. Nommez-la **backup** et définissez son type sur **xfs**. Démarrerez la partition au secteur 2048.

```
[root@serverb ~]# parted /dev/vdb mkpart backup xfs 2048s 2GB  
Information: You may need to update /etc/fstab.
```

- 1.7. Vérifiez que la nouvelle partition a été correctement créée.

```
[root@serverb ~]# parted /dev/vdb print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2000MB	1999MB			backup

- 1.8. Exécutez la commande **udevadm settle**. Cette commande attend que le système détecte la nouvelle partition et crée le fichier de périphérique **/dev/vdb1**. Elle ne revient que lorsque l'opération est terminée.

```
[root@serverb ~]# udevadm settle
[root@serverb ~]#
```

2. Formatez la partition de 2 Go avec un système de fichiers XFS et montez-la de manière persistante dans **/backup**.

- 2.1. Utilisez la commande **mkfs.xfs** pour formater la partition **/dev/vbd1**.

```
[root@serverb ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=121984 blks
                                =                      sectsz=512  attr=2, projid32bit=1
                                =                      crc=1      finobt=1, sparse=1, rmapbt=0
                                =                      reflink=1
data     =                      bsize=4096   blocks=487936, imaxpct=25
        =                      sunit=0      swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=2560, version=2
        =                      sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
```

- 2.2. Créez le point de montage **/backup**.

```
[root@serverb ~]# mkdir /backup
[root@serverb ~]#
```

- 2.3. Avant d'ajouter le nouveau système de fichiers à **/etc/fstab**, récupérez son UUID.

```
[root@serverb ~]# lsblk --fs /dev/vdb1
NAME FSTYPE LABEL UUID                                     MOUNTPOINT
vdb1 xfs      a3665c6b-4bfb-49b6-a528-74e268b058dd
```

L'UUID est probablement différent sur votre système.

- 2.4. Modifiez **/etc/fstab** et définissez le nouveau système de fichiers.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4bfb-49b6-a528-74e268b058dd  /backup  xfs  defaults  0 0
```

- 2.5. Forcez systemd à relire le fichier **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]#
```

**CHAPITRE 6 |** Gestion du stockage de base

- 2.6. Montez manuellement **/backup** pour vérifier votre travail. Confirmez que le montage est réussi.

```
[root@serverb ~]# mount /backup
[root@serverb ~]# mount | grep /backup
/dev/vdb1 on /backup type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

3. Sur ce même nouveau disque, créez deux partitions GPT de 512 Mo nommées **swap1** et **swap2**. Une taille entre 460 et 564 Mo est acceptable. Définissez le type de système de fichiers approprié sur ces partitions pour héberger des espaces d'échange.

- 3.1. Récupérez la position de fin de la première partition en affichant la table de partitions actuelle sur **/dev/vdb**. À l'étape suivante, utilisez cette valeur comme début de la partition **swap1**.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  2000MB  1999MB  xfs          backup
```

- 3.2. Créez la première partition de 512 Mo nommée **swap1**. Définissez son type sur **linux-swap**. Utilisez la position de fin de la première partition comme point de départ. La position de fin est 2 000 Mo + 512 Mo = 2 512 Mo

```
[root@serverb ~]# parted /dev/vdb mkpart swap1 linux-swap 2000MB 2512M
Information: You may need to update /etc/fstab.
```

- 3.3. Créez la seconde partition de 512 Mo appelée **swap2**. Définissez son type sur **linux-swap**. Utilisez la position de fin de la première partition comme point de départ : **2512M**. La position de fin est 2 512 Mo + 512 Mo = 3 024 Mo

```
[root@serverb ~]# parted /dev/vdb mkpart swap2 linux-swap 2512M 3024M
Information: You may need to update /etc/fstab.
```

- 3.4. Affichez la table de partitions pour vérifier votre travail.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  2000MB  1999MB  xfs          backup
 2      2000MB   2512MB  513MB   swap1        swap
 3      2512MB   3024MB  512MB   swap2        swap
```

**CHAPITRE 6 |** Gestion du stockage de base

- 3.5. Exécutez la commande **udevadm settle**. Cette commande attend que le système enregistre les nouvelles partitions et crée les fichiers de périphérique.

```
[root@serverb ~]# udevadm settle  
[root@serverb ~]#
```

4. Initialisez les deux partitions de 512 Mio en tant qu'espaces d'échange et configurez-les de sorte qu'elles soient activées au démarrage. Définissez l'espace d'échange **swap2** comme étant prioritaire sur l'autre.

- 4.1. Utilisez la commande **mkswap** pour initialiser les partitions d'échange.

```
[root@serverb ~]# mkswap /dev/vdb2  
Setting up swapspace version 1, size = 489 MiB (512749568 bytes)  
no label, UUID=87976166-4697-47b7-86d1-73a02f0fc803  
[root@serverb ~]# mkswap /dev/vdb3  
Setting up swapspace version 1, size = 488 MiB (511700992 bytes)  
no label, UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942
```

Prenez note des UUID des deux espaces d'échange. Vous allez utiliser ces informations au cours de la prochaine étape. Si vous ne pouvez plus voir la sortie **mkswap**, utilisez la commande **lsblk --fs** pour récupérer les UUID.

- 4.2. Modifiez **/etc/fstab** et définissez les nouveaux espaces d'échange. Pour définir l'espace d'échange sur la partition **swap2** plutôt que sur **swap1**, accordez-lui une priorité plus élevée avec l'option **pri**.

```
[root@serverb ~]# vim /etc/fstab  
...output omitted...  
UUID=a3665c6b-4fbf-49b6-a528-74e268b058dd  /backup xfs  defaults  0 0  
UUID=87976166-4697-47b7-86d1-73a02f0fc803  swap    swap  pri=10  0 0  
UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942  swap    swap  pri=20  0 0
```

- 4.3. Forcez **systemd** à relire le fichier **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload  
[root@serverb ~]#
```

- 4.4. Utilisez la commande **swapon -a** pour activer les nouveaux espaces d'échange.

Utilisez la commande **swapon --show** pour vérifier que les espaces d'échange sont correctement activés.

```
[root@serverb ~]# swapon -a  
[root@serverb ~]# swapon --show  
NAME      TYPE      SIZE USED PRI  
/dev/vdb2 partition 489M   0B   10  
/dev/vdb3 partition 488M   0B   20
```

5. Pour vérifier votre travail, redémarrez **serverb**. Vérifiez que le système monte automatiquement la première partition dans **/backup**. Vérifiez également que le système active les deux espaces d'échange.

Lorsque vous avez terminé, déconnectez-vous de **serverb**.

### 5.1. Redémarrez serverb.

```
[root@serverb ~]# systemctl reboot
[root@serverb ~]#
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

### 5.2. Attendez quelques minutes que serverb redémarre, puis connectez-vous en tant qu'utilisateur student.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

### 5.3. Vérifiez que le système se monte automatiquement /dev/vdb1 dans /backup.

```
[student@serverb ~]$ mount | grep /backup
/dev/vdb1 on /backup type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

### 5.4. Utilisez la commande **swapon --show** pour vérifier que le système active les deux espaces d'échange.

```
[student@serverb ~]$ swapon --show
NAME      TYPE      SIZE USED PRI
/dev/vdb2 partition 489M   0B   10
/dev/vdb3 partition 488M   0B   20
```

### 5.5. Déconnectez-vous de serverb.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

## Évaluation

À partir de workstation, exécutez le script **lab storage-review grade** pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab storage-review grade
```

## Fin

Sur workstation, exécutez le script **lab storage-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab storage-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Vous utilisez la commande **parted** pour ajouter, modifier et supprimer des partitions sur les disques selon le schéma de partitionnement MBR ou GPT.
- Vous utilisez la commande **mkfs.xfs** pour créer des systèmes de fichiers XFS sur des partitions de disque.
- Vous devez ajouter des commandes de montage de système de fichiers à **/etc/fstab** pour rendre ces montages persistants.
- Vous utilisez la commande **mkswap** pour initialiser les espaces d'échange.

## CHAPITRE 7

# GESTION DES VOLUMES LOGIQUES

### PROJET

Créer et gérer les volumes logiques contenant des systèmes de fichiers et des espaces d'échange à partir de la ligne de commande.

### OBJECTIFS

- Créer et gérer des volumes logiques à partir de périphériques de stockage, puis les formater à l'aide de systèmes de fichiers ou les préparer avec des espaces d'échange.
- Ajouter et supprimer le stockage attribué aux groupes de volumes et augmenter de manière non destructive la taille d'un volume logique formaté avec un système de fichiers.

### SECTIONS

- Création de volumes logiques (et exercice guidé)
- Extension de volumes logiques (et exercice guidé)

### ATELIER

Gestion des volumes logiques

# CRÉATION DE VOLUMES LOGIQUES

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Décrire les différents concepts et composants de gestion des volumes logiques.
- Mettre en œuvre des stockages LVM.
- Afficher les informations sur les composants LVM.

## CONCEPTS RELATIFS À LA GESTION DES VOLUMES LOGIQUES (LVM)

Les volumes logiques et le gestionnaire de volumes logiques facilitent la gestion de l'espace disque. Si un système de fichiers qui héberge un volume logique nécessite davantage d'espace, l'espace disponible de son groupe de volumes peut être alloué à son volume logique. Le système de fichiers peut alors être redimensionné. Si un disque commence à donner des signes de faiblesse, un disque de rechange peut être déclaré comme volume physique dans le groupe de volumes, et les étendues du volume logique peuvent alors être déplacées vers le nouveau disque.

### Définitions LVM

#### Périphériques physiques

Les périphériques physiques sont les périphériques de stockage utilisés pour conserver les données stockées dans un volume logique. Il s'agit de périphériques en mode bloc qui peuvent prendre la forme de partitions de disque, de disques entiers, de matrices RAID ou de disques SAN. Un périphérique doit être initialisé en tant que volume physique LVM pour être utilisable avec LVM. La totalité du périphérique sera utilisée en tant que volume physique.

#### Volumes physiques (PV, Physical Volumes)

Vous devez initialiser un périphérique en tant que volume physique avant de l'utiliser dans un système LVM. Les outils LVM segmentent les volumes physiques sous la forme d'*étendues physiques (PE, Physical Extents)* ; ces étendues sont de petits fragments de données qui constituent le plus petit bloc de stockage d'un volume physique.

#### Groupes de volumes (VG, Volume Groups)

Les groupes de volumes sont des pools de stockage constitués d'un ou de plusieurs volumes physiques. C'est l'équivalent fonctionnel d'un disque entier dans le stockage de base. Un PV ne peut être alloué qu'à un seul VG. Un VG peut se composer d'espace inutilisé et d'un nombre quelconque de volumes logiques.

#### Volumes logiques (LV, Logical Volumes)

Les volumes logiques sont créés à partir d'étendues physiques libres dans un groupe de volumes et fournissent le périphérique de « stockage » utilisé par les applications, les utilisateurs et le système d'exploitation. Les LV sont un groupe d'*étendues logiques (LE, Logical Extents)* qui sont mises en correspondance avec des extensions physiques, la plus petite unité de stockage d'un PV. Par défaut, chaque LE est mis en correspondance avec une seule PE. La définition d'options de LV spécifiques modifie cette mise en correspondance ; par exemple, le mode miroir entraîne la mise en correspondance de chaque LE avec deux PE.

## MISE EN ŒUVRE D'UNITÉS DE STOCKAGE LVM

La création d'un espace de stockage LVM nécessite plusieurs étapes. La première étape consiste à déterminer les périphériques physiques à utiliser. Une fois un ensemble de périphériques appropriés assemblés, ces derniers sont initialisés en tant que volumes physiques, de sorte qu'ils soient reconnus comme appartenant à LVM. Les volumes physiques sont ensuite combinés en un groupe de volumes. Cela crée un pool d'espace disque à partir duquel des volumes logiques peuvent être alloués. Les volumes logiques créés à partir de l'espace disponible dans un groupe de volumes peuvent être formatés avec un système de fichiers, activés en tant qu'espace d'échange et montés ou activés de manière persistante.

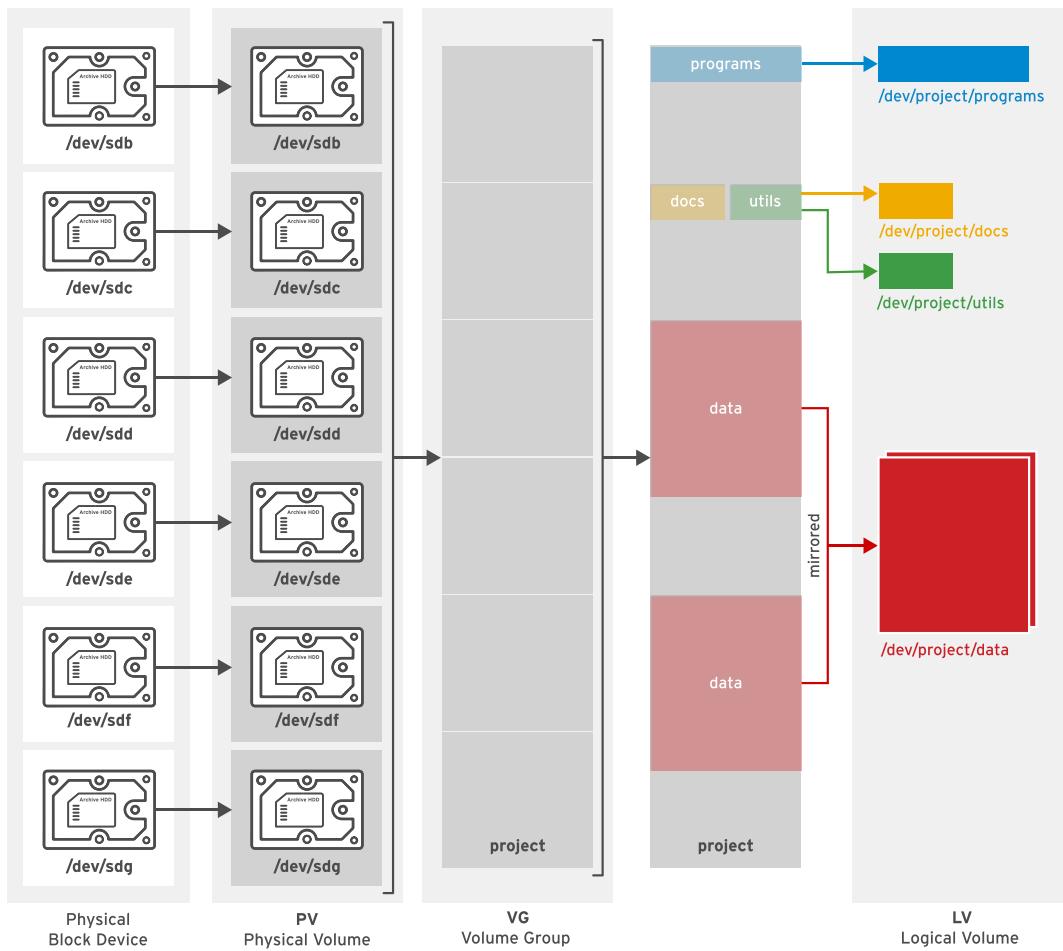


Figure 7.1: Composants du gestionnaire de volumes logiques

LVM offre un ensemble complet d'outils en ligne de commande pour mettre en œuvre et gérer des stockages LVM. On peut utiliser ces outils en ligne de commande dans des scripts, ce qui confirme qu'ils sont adaptés à l'automatisation.



### IMPORTANT

Les exemples qui suivent utilisent le périphérique **vdb** et ses partitions pour illustrer les commandes LVM. Dans la pratique, ces exemples devraient utiliser les bons périphériques associés pour le disque et les partitions de disque effectivement utilisés par le système. Utilisez les commandes **lsblk**, **blkid** ou **cat /proc/partitions** pour identifier les périphériques sur votre système.

## Création d'un volume logique

Pour créer un volume logique, procédez comme suit :

### Préparez le périphérique physique.

Utilisez **parted**, **gdisk** ou **fdisk** pour créer une partition à utiliser avec LVM. Choisissez toujours le type de partition **Linux LVM** pour les partitions LVM, et **0x8e** pour les partitions MBR. Si nécessaire, utilisez la commande **partprobe** pour enregistrer la nouvelle partition auprès du noyau.

Vous pouvez aussi utiliser un disque entier, une matrice RAID ou un disque SAN.

La préparation d'un périphérique physique n'est requise que si aucun autre périphérique n'a été préparé et qu'un nouveau volume physique est requis pour la création ou l'extension d'un groupe de volumes.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1MiB 769MiB
[root@host ~]# parted -s /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted -s /dev/vdb set 1 lvm on
[root@host ~]# parted -s /dev/vdb set 2 lvm on
```

### Créez un volume physique.

Utilisez **pvcreate** pour étiqueter la partition (ou un autre périphérique physique) en tant que volume physique. La commande **pvcreate** divise le volume physique en étendues physiques (PE) de taille fixe, par exemple, des blocs de 4 Mio. Vous pouvez étiqueter plusieurs périphériques à la fois en passant les noms de périphérique en arguments dans la commande **pvcreate**, séparés par un espace.

```
[root@host ~]# pvcreate /dev/vdb2 /dev/vdb1
```

Cette opération étiquette les périphériques **/dev/vdb2** et **/dev/vdb1** comme PV prêts à être alloués à un groupe de volumes.

La création d'un PV n'est requise que s'il n'existe aucun PV libre pour créer ou étendre un VG.

### Créez un groupe de volumes.

Utilisez **vgcreate** pour combiner un ou plusieurs volumes physiques en un groupe de volumes. Un groupe de volumes est l'équivalent fonctionnel d'un disque dur. vous allez créer des volumes logiques à partir du pool d'extensions physiques libres du groupe de volumes.

La ligne de commande **vgcreate** se compose d'un nom de groupe de volumes suivi d'un ou de plusieurs volumes physiques à allouer à ce groupe de volumes.

```
[root@host ~]# vgcreate vg01 /dev/vdb2 /dev/vdb1
```

Cette opération crée un VG appelé **vg01** dont la taille, exprimée en unités PE, est égale à la somme des deux PV **/dev/vdb2** et **/dev/vdb1**.

Un VG ne doit être créé que s'il n'en existe pas déjà. Il est possible de créer d'autres VG à des fins administratives pour gérer l'utilisation des PV et des LV. Dans le cas contraire, les VG peuvent être étendus de façon à prendre en charge les nouveaux LV en cas de besoin.

## Créez un volume logique.

Utilisez **lvcreate** pour créer un volume logique à partir des étendues physiques disponibles dans un groupe de volumes. La commande **lvcreate** comprend au minimum l'option **-n** pour définir le nom du LV, l'option **-L** pour définir la taille du LV en octets ou l'option **-l** pour définir la taille du LV en étendues, et le nom du groupe de volumes hébergeant ce volume logique.

```
[root@host ~]# lvcreate -n lv01 -L 700M vg01
```

Cela crée un LV appelé **lv01** d'une taille de 700 Mio, dans le VG **vg01**. Cette commande échoue si le groupe de volumes ne dispose pas d'un nombre suffisant d'extensions physiques libres pour la taille demandée. Notez également que la taille est arrondie au facteur de la taille de l'étendue physique si la taille ne peut pas correspondre exactement.

Vous pouvez spécifier la taille en utilisant l'option **-L**, qui attend des tailles en octets, mébiocets (mégaoctets binaires, 1 048 576 octets), gibiocets (gigaoctets binaires) ou similaire. Sinon, vous pouvez utiliser l'option **-l** qui attend des tailles spécifiées en un nombre d'extensions physiques.

*La liste suivante fournit des exemples de création de volumes logiques :*

- **lvcreate -L 128M** : règle la taille du volume logique à 128 Mio exactement.
- **lvcreate -l 128** : règle la taille du volume logique sur 128 étendues exactement. Le nombre total d'octets dépend de la taille du bloc d'étendues physiques sur le volume physique sous-jacent.



### IMPORTANT

Les différents outils affichent le nom de volume logique en utilisant soit le nom traditionnel, **/dev/nomvg/nomlv**, soit le nom du mappeur de périphérique du noyau, **/dev/mapper/nomvg/nomlv**.

## Ajoutez le système de fichiers.

Utilisez **mkfs** pour créer un système de fichiers **XFS** sur le nouveau volume logique. Vous pouvez aussi créer un système de fichiers en fonction de votre système de fichiers favori, par exemple **ext4**.

```
[root@host ~]# mkfs -t xfs /dev/vg01/lv01
```

Pour que le système de fichiers reste disponible d'un redémarrage à l'autre , effectuez les étapes suivantes :

- Utilisez **mkdir** pour créer un point de montage.

```
[root@host ~]# mkdir /mnt/data
```

- Ajoutez une entrée au fichier **/etc/fstab**:

```
/dev/vg01/lv01 /mnt/data xfs defaults 1 2
```

**NOTE**

Le montage d'un volume logique par nom revient au montage par UUID, car LVM recherche ses volumes physiques en fonction d'un UUID, même si vous les avez initialement ajoutés au groupe de volumes par nom.

- Exécutez **mount /mnt/data** pour monter le système de fichiers que vous venez d'ajouter dans **/etc/fstab**.

```
[root@host ~]# mount /mnt/data
```

## Suppression d'un volume logique

Pour supprimer *tous* les composants d'un volume logique, procédez comme suit :

### Préparez le système de fichiers.

Déplacez toutes les données à conserver vers un autre système de fichiers. Utilisez **umount** pour démonter le système de fichiers, puis supprimez toute entrée **/etc/fstab** associée à ce système de fichiers.

```
[root@host ~]# umount /mnt/data
```

**MISE EN GARDE**

La suppression d'un volume logique entraîne la destruction de toutes les données stockées sur le volume logique. Sauvegardez ou déplacez vos données *avant* de supprimer le volume logique.

### Supprimez le volume logique.

Utilisez **lvremove DEVICE\_NAME** pour supprimer un volume logique devenu inutile.

```
[root@host ~]# lvremove /dev/vg01/lv01
```

Démontez le système de fichiers du LV avant d'exécuter cette commande. La commande demande confirmation avant de supprimer le LV.

Les étendues physiques du LV sont libérées et rendues disponibles pour une assignation à des LV existants ou nouvellement créés dans le groupe de volumes.

### Supprimez le groupe de volumes.

Utilisez **vgremove VG\_NAME** pour supprimer un groupe de volumes devenu inutile.

```
[root@host ~]# vgremove vg01
```

Les volumes physiques du VG sont alors libérés et rendus disponibles pour assignation à des VG existants ou nouvellement créés sur le système.

### Supprimez les volumes physiques.

Utilisez **pvremove** pour supprimer les volumes physiques devenus inutiles. Utilisez une liste de PV séparés par des espaces pour supprimer plusieurs périphériques PV à la fois. Cette commande supprime les métadonnées PV de la partition (ou du disque). La partition peut maintenant être réaffectée ou reformatée.

```
[root@host ~]# pvremove /dev/vdb2 /dev/vdb1
```

## CONSULTATION DES INFORMATIONS D'ÉTAT LVM

### Volumes physiques

Utilisez **pvdisplay** pour visualiser les informations relatives aux volumes physiques. Pour lister des informations sur tous les volumes physiques, utilisez la commande sans argument. Pour lister les informations relatives à un volume physique spécifique, transmettez ce nom de périphérique à la commande.

```
[root@host ~]# pvdisplay /dev/vdb1
--- Physical volume ---
PV Name              /dev/vdb1          ①
VG Name              vg01               ②
PV Size              768.00 MiB / not usable 4.00 MiB ③
Allocatable          yes
PE Size              4.00 MiB          ④
Total PE             191
Free PE              16                ⑤
Allocated PE         175
PV UUID              JWzDpn-LG3e-n2oi-9Etd-VT2H-PMem-1ZXwP1
```

- ① **PV Name** est mis en correspondance avec le nom du périphérique.
- ② **VG Name** indique le groupe de volumes auquel le PV est alloué.
- ③ **PV Size** affiche la taille physique du PV, espace inutilisable inclus.
- ④ **PE Size** est la taille d'étendue physique, qui constitue la plus petite taille pouvant être allouée à un volume logique.

Il s'agit également du coefficient multiplicateur lors du calcul de la taille de toute valeur exprimée en unités PE, comme *Free PE*. Par exemple, 26 PE x 4 Mio (la *taille de PE*) équivaut à 104 Mio d'espace libre. La taille d'un volume logique est arrondie à un multiple d'unités PE.

- LVM définit automatiquement la taille des PE, bien qu'il soit possible de la spécifier.
- ⑤ **Free PE** indique le nombre d'unités PE disponibles pour affectation à de nouveaux volumes logiques.

### Groupes de volumes

Utilisez **vgdisplay** pour afficher les informations relatives aux groupes de volumes. Pour lister des informations sur tous les groupes de volumes, utilisez la commande sans argument. Pour lister les informations relatives à un groupe de volumes spécifique, transmettez ce nom de VG à la commande.

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name              vg01          ①
System ID            lvm2
Format
```

**CHAPITRE 7 |** Gestion des volumes logiques

```

Metadata Areas      2
Metadata Sequence No 2
VG Access          read/write
VG Status           resizable
MAX LV              0
Cur LV              1
Open LV              1
Max PV              0
Cur PV              2
Act PV              2
VG Size             1016.00 MiB    ②
PE Size              4.00 MiB
Total PE             254            ③
Alloc PE / Size     175 / 700.00 MiB
Free PE / Size      79 / 316.00 MiB ④
VG UUID             3snNw3-CF71-CcYG-Llk1-p6EY-rHEv-xfUSez

```

- ① **VG Name** est le nom du groupe de volumes.
- ② **VG Size** indique la taille totale du pool de stockage disponible pour allocation aux volumes logiques.
- ③ **Total PE** correspond à la taille totale exprimée en unités PE.
- ④ **Free PE / Size** montre l'espace libre dans le VG pour affectation à de nouveaux LV ou pour une extension de LV existants.

## Volumes logiques

Utilisez **lvdisplay** pour visualiser les informations relatives aux volumes logiques. Si vous ne fournissez aucun argument à la commande, elle affiche les informations sur tous les LV ; si vous fournissez le nom d'un périphérique LV en argument, la commande affiche les informations propres à ce périphérique.

```

[root@host ~]# lvdisplay /dev/vg01/lv01
--- Logical volume ---
LV Path              /dev/vg01/lv01    ①
LV Name              lv01
VG Name              vg01            ②
LV UUID              5IyRea-W8Zw-xLHK-3h2a-IuVN-YaeZ-i3IRrN
LV Write Access      read/write
LV Creation host, time host.lab.example.com, 2019-03-28 17:17:47 -0400
LV Status            available
# open               1
LV Size              700 MiB        ③
Current LE           175            ④
Segments             1
Allocation           inherit
Read ahead sectors   auto
- current set to    256
Block device         252:0

```

- ① **LV Path** indique le nom de périphérique du volume logique.

Certains outils peuvent spécifier le nom de périphérique sous la forme **/dev/mapper/vgname-lvname**, les deux représentant le même LV.

- ② **VG Name** indique le groupe de volumes à partir duquel le LV est alloué.

- ③ **LV Size** affiche la taille totale du LV. Utilisez des outils pour systèmes de fichiers afin de déterminer l'espace libre et l'espace utilisé pour le stockage de données.
- ④ **Current LE** affiche le nombre d'étendues logiques utilisées par ce LV. Une LE est généralement mise en correspondance avec une étendue physique du VG, et par conséquent avec le volume physique.



## RÉFÉRENCES

Pages de manuel **lvm(8)**, **pvccreate(8)**, **vgcreate(8)**, **lvcreate(8)**,  
**pvremove(8)**, **vgremove(8)**, **lvremove(8)**, **pvdisplay(8)**, **vgdisplay(8)**,  
**lvdisplay(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)** et **mkfs(8)**

## ► EXERCICE GUIDÉ

# CRÉATION DE VOLUMES LOGIQUES

Dans le cadre de cet atelier, vous allez créer un volume physique, un groupe de volumes, un volume logique et un système de fichiers XFS. Vous monterez également le système de fichiers du volume logique de façon persistante.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer des volumes physiques, des groupes de volumes et des volumes logiques avec les outils LVM.
- Créer des systèmes de fichiers sur des volumes logiques et les monter de manière persistante.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab lvm-creating start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle vérifie également que le stockage est disponible et que les paquetages logiciels appropriés sont installés.

```
[student@workstation ~]$ lab lvm-creating start
```

- 1. Utilisez la commande `ssh` pour vous connecter à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Utilisez la commande `sudo -i` pour basculer vers l'utilisateur `root`. Le mot de passe de l'utilisateur `student` est `student`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Créez les ressources physiques.

- 3.1. Utilisez `parted` pour créer deux partitions de 256 Mio chacune et attribuez-leur le type Linux LVM.

```
[root@servera ~]# parted -s /dev/vdb mklabel gpt
[root@servera ~]# parted -s /dev/vdb mkpart primary 1MiB 257MiB
[root@servera ~]# parted -s /dev/vdb set 1 lvm on
[root@servera ~]# parted -s /dev/vdb mkpart primary 258MiB 514MiB
[root@servera ~]# parted -s /dev/vdb set 2 lvm on
```

3.2. Utilisez **udevadm settle** pour que le système enregistre les nouvelles partitions.

```
[root@servera ~]# udevadm settle
```

► 4. Utilisez **pvcreate** pour ajouter les deux nouvelles partitions en tant que PV.

```
[root@servera ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
```

► 5. Utilisez **vgcreate** pour créer un VG appelé **servera\_01\_vg** à partir des deux PV.

```
[root@servera ~]# vgcreate servera_01_vg /dev/vdb1 /dev/vdb2
Volume group "servera_01_vg" successfully created
```

► 6. Utilisez **lvcreate** pour créer un LV de 400 Mio appelé **servera\_01\_lv** à partir du VG **servera\_01\_vg**.

```
[root@servera ~]# lvcreate -n servera_01_lv -L 400M servera_01_vg
Logical volume "servera_01_lv" created.
```

Cela crée un périphérique nommé **/dev/servera\_01\_vg/servera\_01\_lv**, qui est toutefois dépourvu de système de fichiers.

► 7. Ajoutez un système de fichiers persistant.

7.1. Ajoutez un système de fichiers XFS sur le LV **servera\_01\_lv** avec la commande **mkfs**.

```
[root@servera ~]# mkfs -t xfs /dev/servera_01_vg/servera_01_lv
...output omitted...
```

7.2. Créez un point de montage dans **/data**.

```
[root@servera ~]# mkdir /data
```

7.3. Ajoutez la ligne suivante à la fin de **/etc/fstab** sur **servera**:

```
/dev/servera_01_vg/servera_01_lv    /data    xfs    defaults    1 2
```

7.4. Utilisez **systemctl daemon-reload** pour mettre à jour **systemd** avec la nouvelle configuration **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

- 7.5. Vérifiez l'entrée **/etc/fstab** et montez le nouveau périphérique LV **servera\_01\_lv** avec la commande **mount**.

```
[root@servera ~]# mount /data
```

► 8. Testez et passez en revue votre travail.

- 8.1. Pour tester votre travail, copiez certains fichiers dans le répertoire **/data** et comptabilisez les fichiers copiés.

```
[root@servera ~]# cp -a /etc/*.* /data
[root@servera ~]# ls /data | wc -l
34
```

Vous vérifierez que vous disposez toujours du même nombre de fichiers au cours du prochain exercice guidé.

- 8.2. **parted /dev/vdb print** liste les partitions en place sur **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
 1      1049kB  269MB  268MB   primary      lvm
 2      271MB   539MB  268MB   primary      lvm
```

Remarquez la colonne **Number** qui contient les valeurs **1** et **2**. Celles-ci correspondent à **/dev/vdb1** et **/dev/vdb2**, respectivement. Notez également la colonne **Flags** qui indique le type de partition.

- 8.3. La commande **pvdisplay** présente les informations relatives à chacun des volumes physiques. Vous pouvez indiquer le nom du périphérique pour limiter les détails à un PV spécifique.

```
[root@servera ~]# pvdisplay /dev/vdb2
--- Physical volume ---
PV Name           /dev/vdb2
VG Name           servera_01_vg
PV Size          256.00 MiB / not usable 4.00 MiB
Allocatable       yes
PE Size          4.00 MiB
Total PE         63
Free PE          26
Allocated PE     37
PV UUID          2z0Cf3-99YI-w9ny-a1EW-wWhL-S8RJ-M2rfZk
```

**CHAPITRE 7 |** Gestion des volumes logiques

Cela montre que le PV est alloué au VG **servera\_01\_vg**, qu'il fait 256 Mio (dont 4 Mio non utilisables) et que la taille d'étendue physique (**PE Size**) est de 4 Mio (la plus petite taille pouvant être allouée à un LV).

Il existe 63 PE, dont 26 pourront être allouées à des LV par la suite et 37 sont actuellement allouées à des LV. Les valeurs en Mio correspondantes sont les suivantes :

- Total : 252 Mio (63 PE x 4 Mio) ; rappelez-vous que 4 Mio sont inutilisables.
- Libre : 104 Mio (26 PE x 4 Mio)
- Alloué : 148 Mio (37 PE x 4 Mio)

8.4. **vgdisplay vgname** affiche les informations relatives au groupe de volumes nommé vgname.

```
[root@servera ~]# vgdisplay servera_01_vg
```

Vérifiez les valeurs suivantes :

- **VG Size : 504,00 Mio.**
- **Total PE : 126.**
- **Alloc PE / Size : 100 / 400,00 Mio.**
- **Free PE / Size : 26 / 104,00 Mio.**

8.5. La commande **lvdisplay /dev/vgname/lvname** affiche les informations relatives au volume logique nommé lvname.

```
[root@servera ~]# lvdisplay /dev/servera_01_vg/servera_01_lv
```

Notez la présence des valeurs **LV Path**, **LV Name**, **VG Name**, **LV Status**, **LV Size** et **Current LE** (étendues logiques mises en correspondance avec les étendues physiques).

8.6. La commande **mount** affiche tous les périphériques montés et toutes les options de montage. **/dev/servera\_01\_vg/servera\_01\_lv** doit y figurer.

**NOTE**

De nombreux outils renvoient plutôt le nom du mappeur de périphérique, **/dev/mapper/servera\_01\_vg-servera\_01\_lv**. Il s'agit du même volume logique.

```
[root@servera ~]# mount
```

Vous devriez voir (probablement sur la dernière ligne) **/dev/mapper/servera\_01\_vg-servera\_01\_lv** monté sur **/data**, ainsi que les informations de montage associées.

8.7. La commande **df -h** affiche la place disponible sur les disques sous une forme facilement lisible. Vous pouvez préciser le point de montage pour limiter les détails à ce seul système de fichiers, si vous le souhaitez.

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  395M   24M  372M   6% /data
```

Compte tenu des métadonnées du système de fichiers, ces valeurs sont conformes aux attentes.

► **9.** Déconnectez-vous de servera.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez le script **lab lvm-creating finish** pour mettre fin à l'exercice. Ce script supprime toutes les unités de stockage installées sur servera pendant l'exercice.

```
[student@workstation ~]$ lab lvm-creating finish
```

L'exercice guidé est maintenant terminé.

# EXTENSION DES VOLUMES LOGIQUES

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Étendre un groupe de volumes (VG) à l'aide de **pvccreate** et **vgextend**, et utiliser la commande **vgdisplay** pour vérifier les résultats.
- Réduire un VG à l'aide de **pvmove** et **vgreduce**.
- Étendre les volumes logiques (LV) à l'aide de **lvextend**.
- Redimensionner les systèmes de fichiers **XFS** avec **xfs\_growfs**.
- Redimensionner les systèmes de fichiers **ext4** avec **resize2fs**.

## EXTENSION ET RÉDUCTION D'UN GROUPE DE VOLUMES

Vous pouvez augmenter l'espace disque d'un groupe de volumes en y ajoutant des volumes physiques supplémentaires. C'est ce que l'on appelle étendre *un groupe de volumes*. Vous pouvez ensuite affecter les nouvelles étendues physiques des volumes physiques supplémentaires à des volumes logiques.

Vous pouvez supprimer les volumes physiques inutilisés d'un groupe de volumes. C'est ce que l'on appelle réduire *le groupe de volumes*. Au préalable, utilisez la commande **pvmove** pour déplacer les données des étendues d'un volume physique vers les étendues d'autres volumes physiques du groupe de volumes. Ainsi, on peut ajouter un nouveau disque à un groupe de volumes existant, déplacer des données d'un disque plus ancien ou plus lent vers un nouveau disque et supprimer l'ancien disque du groupe de volumes. Vous pouvez effectuer ces actions pendant que les volumes logiques du groupe de volumes sont en cours d'utilisation.



### IMPORTANT

Les exemples qui suivent utilisent le périphérique **vdb** et ses partitions pour illustrer les commandes LVM. En pratique, utilisez les périphériques appropriés pour le disque et les partitions de disque sur votre propre système.

### Extension d'un groupe de volumes

Pour étendre un groupe de volumes, procédez comme suit :

#### Préparez le périphérique physique et créer le volume physique.

Comme dans le cadre de la création d'un groupe de volumes, vous devez créer et préparer une partition afin de l'utiliser en tant que volume physique LVM, si cela n'a pas encore été préparé.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1027MiB 1539MiB
[root@host ~]# parted -s /dev/vdb set 3 lvm on
[root@host ~]# pvccreate /dev/vdb3
```

## CHAPITRE 7 | Gestion des volumes logiques

La création d'un PV n'est requise que s'il n'y a aucun PV libre pour étendre le VG.

### Étendez le groupe de volumes.

Utilisez **vgextend** pour ajouter le nouveau volume physique au groupe de volumes. Utilisez le nom du VG et le nom du périphérique PV comme arguments de **vgextend**.

```
[root@host ~]# vgextend vg01 /dev/vdb3
```

Cela étend le VG vg01 de la taille du PV **/dev/vdb3**.

### Vérifiez que le nouvel espace est disponible.

Utilisez **vgdisplay** pour vous assurer que les étendues physiques supplémentaires sont disponibles. Examinez la valeur de la chaîne **Free PE / Size** dans la sortie. Elle doit être différente de zéro.

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name           vg01
...output omitted...
Free  PE / Size   178 / 712.00 MiB
...output omitted...
```

## Réduction d'un groupe de volumes

Pour réduire un groupe de volumes, procédez comme suit :

### Déplacez les étendues physiques.

Utilisez **pvmove PV\_DEVICE\_NAME** pour déplacer n'importe quelle extension physique du volume physique que vous souhaitez supprimer vers d'autres volumes physiques du groupe de volumes. Les autres volumes physiques doivent avoir un nombre suffisant d'étendues libres pour permettre ce déplacement. Cela n'est possible que s'il y a suffisamment d'étendues libres dans le VG et qu'elles proviennent toutes d'autres PV.

```
[root@host ~]# pvmove /dev/vdb3
```

Cette commande déplace les PE de **/dev/vdb3** vers d'autres PV disposant de PE libres dans le même VG.



### MISE EN GARDE

Avant d'utiliser **pvmove**, sauvegardez les données stockées sur tous les volumes logiques du groupe de volumes. Toute coupure de courant au cours de l'opération risque de laisser le groupe de volumes dans un état incohérent. Il pourrait en résulter une perte de données sur les volumes logiques du groupe de volumes.

### Réduisez le groupe de volumes.

Utilisez **vgreduce VG\_NAME PV\_DEVICE\_NAME** pour supprimer un volume physique du groupe de volumes.

```
[root@host ~]# vgreduce vg01 /dev/vdb3
```

Cette action supprime le PV **/dev/vdb3** du VG **vg01**. Ce dernier peut désormais être ajouté à un autre VG. La commande **pvremove** permet également d'arrêter d'utiliser le périphérique en tant que PV.

## EXTENSION D'UN VOLUME LOGIQUE ET D'UN SYSTÈME DE FICHIERS XFS

L'un des avantages des volumes logiques réside dans la possibilité d'augmenter leur taille sans interruption de service. Les étendues physiques libres d'un groupe de volumes peuvent être ajoutées à un volume logique pour accroître sa capacité, qui peut à son tour servir à étendre le système de fichiers qu'il contient.

### Extension d'un volume logique

Pour étendre un volume logique, procédez comme suit :

#### Vérifiez que le groupe de volumes contient de l'espace disponible.

Utilisez **vgdisplay** pour vérifier que les extensions physiques disponibles sont suffisantes.

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name           vg01
...output omitted...
Free PE / Size   178 / 712.00 MiB
...output omitted...
```

Examinez la valeur de la chaîne **Free PE / Size** dans la sortie. Vérifiez que le groupe de volumes dispose de suffisamment d'espace libre pour l'extension LV. Si l'espace disponible est insuffisant, étendez le groupe de volumes de manière appropriée. Consultez la section intitulée « Extension et réduction d'un groupe de volumes ».

#### Étendez le volume logique.

Utilisez **lvextend** *LV\_DEVICE\_NAME* pour augmenter la taille du volume logique à un nouveau seuil.

```
[root@host ~]# lvextend -L +300M /dev/vg01/lv01
```

Cela augmentera la taille du volume logique **lv01** de 300 Mio. Notez le signe plus (+) devant la taille, qui sert à ajouter cette valeur à la taille actuelle ; en l'absence de ce caractère, la valeur définit la taille finale du LV.

Comme avec **lvcreate**, différentes méthodes existent pour spécifier la taille : l'option **-L** attend le nombre d'étendues physiques comme argument. L'option **-L** attend des tailles en octets, mébiocbytes, gibioctets et similaire.

*La liste suivante fournit des exemples d'extension de LV :*

**Exemples d'extension de LV**

COMMANDÉ	RÉSULTATS
<b>lvextend -l 128</b>	Redimensionnez le volume logique à une taille <i>exactement</i> égale à 128 étendues.
<b>lvextend -l +128</b>	Ajoutez 128 étendues à la taille actuelle du volume logique.
<b>lvextend -L 128M</b>	Redimensionnez le volume logique à <i>exactement</i> 128 Mio.
<b>lvextend -L +128M</b>	Ajoutez 128 Mio à la taille actuelle du volume logique.
<b>lvextend -l +50%FREE</b>	Ajoutez au LV 50 pour cent de la place actuellement disponible dans le VG.

**Etendez le système de fichiers.**

Utilisez **xfs\_growfs mountpoint** pour étendre le système de fichiers afin qu'il occupe le LV étendu. Le système de fichiers cible doit être monté lorsque vous utilisez la commande **xfs\_growfs**. Vous pouvez continuer à utiliser le système de fichiers pendant le redimensionnement.

```
[root@host ~]# xfs_growfs /mnt/data
```

**NOTE**

Une erreur courante consiste à exécuter **lvextend** en oubliant d'exécuter **xfs\_growfs**. Au lieu d'exécuter les deux étapes à la suite l'une de l'autre, on peut ajouter l'option **-r** à la commande **lvextend**. Cela redimensionne le système de fichiers après l'extension du LV à l'aide de **fsadm(8)**. Cela fonctionne avec différents systèmes de fichiers.

- Vérifiez la nouvelle taille du système de fichiers monté :

```
df -h /mountpoint.
```

**EXTENSION D'UN VOLUME LOGIQUE ET D'UN SYSTÈME DE FICHIERS EXT4**

Les étapes nécessaires pour étendre un volume logique basé sur **ext4** sont essentiellement les mêmes que pour un LV basé sur **XFS**, à l'exception de l'étape de redimensionnement du système de fichiers. Révision la section intitulée « Extension d'un volume logique et d'un système de fichiers XFS ».

**Vérifiez que le groupe de volumes contient de l'espace disponible.**

Utilisez **vgdisplay VGNAME** pour vérifier que les extensions physiques disponibles pour le groupe de volumes sont suffisantes.

**Etendez le volume logique.**

Utilisez **lvextend -l +extents /dev/vgname/lvname** pour étendre le volume logique */dev/nomvg/nomlv* avec la valeur *extents*.

## Étendez le système de fichiers.

Utilisez **resize2fs /dev/vgname/lvname** pour étendre le système de fichiers afin qu'il occupe le nouveau LV étendu. Le système de fichiers peut être monté et utilisé pendant l'exécution de la commande d'extension. Vous pouvez ajouter l'option **-p** pour contrôler la progression de l'opération de redimensionnement.

```
[root@host ~]# resize2fs /dev/vg01/lv01
```



### NOTE

La différence majeure entre **xfs\_growfs** et **resize2fs** réside dans l'argument transmis pour identifier le système de fichiers. **xfs\_growfs** utilise le point de montage et **resize2fs** utilise le nom du volume logique.

## ÉTENDRE UN VOLUME LOGIQUE ET UN ESPACE D'ÉCHANGE

Les volumes logiques formatés en tant qu'espace d'échange peuvent également être étendus, mais le processus est différent de celui utilisé pour étendre un système de fichiers, tel qu'**ext4** ou **XFS**. Les volumes logiques formatés avec un système de fichiers peuvent être étendus de manière dynamique sans temps d'arrêt. Les volumes logiques formatés avec un espace d'échange doivent être mis hors ligne pour pouvoir être étendus.

### Vérifiez que le groupe de volumes contient de l'espace disponible.

Utilisez **vgdisplay vgname** pour vérifier que les extensions physiques sont disponibles en nombre suffisant.

### Désactivez l'espace d'échange.

Utilisez **swapoff -v /dev/vgname/lvname** pour désactiver l'espace d'échange sur le volume logique.



### MISE EN GARDE

Votre système doit disposer de suffisamment de mémoire libre ou d'espace d'échange pour accepter tout ce qui doit être échangé lorsque l'espace d'échange sur le volume logique est désactivé.

### Etendez le volume logique.

**lvextend -l +extents /dev/vgname/lvname** étend le volume logique */dev/nomvg/nomlv* avec la valeur *extents*.

### Formatez ce volume logique en tant qu'espace d'échange.

**mkswap /dev/vgname/lvname** formate le volume logique entier en tant qu'espace d'échange.

### Activez l'espace d'échange.

Utilisez **swapon -va /dev/vgname/lvname** pour activer l'espace d'échange sur le volume logique.



## RÉFÉRENCES

Pages du manuel **lvm(8)**, **pvccreate(8)**, **pvmove(8)**, **vgdisplay(8)**,  
**vgextend(8)**, **vgreduce(8)**, **vgdisplay(8)**, **vgextend(8)**, **vgreduce(8)**,  
**lvextend(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)**, **xfs\_growfs(8)**  
et **resize2fs(8)** **swapoff(8)** **swapon(8)** **mkswap(8)**

## ► EXERCICE GUIDÉ

# EXTENSION DES VOLUMES LOGIQUES

Dans le cadre de cet atelier, vous allez étendre le volume logique ajouté au cours de l'exercice précédent.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Étendre le groupe de volumes pour inclure un volume physique supplémentaire.
- Redimensionner le volume logique alors que le système de fichiers est encore monté et utilisé.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab lvm-extending start**. Cette commande exécute un script de démarrage qui détermine si l'hôte **servera** est accessible sur le réseau et garantit que le stockage de l'exercice guidé précédent est disponible.

```
[student@workstation ~]$ lab lvm-extending start
```

- 1. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Utilisez la commande **sudo -i** pour basculer vers **root** à l'invite du shell.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Utilisez **vgdisplay** pour déterminer si le VG dispose d'assez de place pour étendre le LV à une taille totale de 700 Mio.

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name          servera_01_vg
System ID
Format          lvm2
...output omitted...
VG Size          504.00 MiB
```

**CHAPITRE 7 |** Gestion des volumes logiques

```

PE Size           4.00 MiB
Total PE         126
Alloc PE / Size 100 / 400.00 MiB
Free  PE / Size 26 / 104.00 MiB
VG UUID          OBBATU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV

```

Seulement 104 Mio sont disponibles (26 PE x étendues de 4 Mio) et vous avez besoin d'au moins 300 Mio pour disposer de 700 Mio au total. Vous devez étendre le VG.

À des fins de comparaison ultérieure, utilisez **df** pour relever l'espace actuellement disponible sur les disques :

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv 395M   24M  372M   6% /data
```

▶ **4.** Créez les ressources physiques.

- 4.1. Utilisez **parted** pour créer une partition supplémentaire de 512 Mio et attribuez-lui le type « Linux LVM ».

```
[root@servera ~]# parted -s /dev/vdb mkpart primary 515MiB 1027MiB
[root@servera ~]# parted -s /dev/vdb set 3 lvm on
```

- 4.2. Utilisez **udevadm settle** pour que le système enregistre la nouvelle partition.

```
[root@servera ~]# udevadm settle
```

▶ **5.** Utilisez **pvccreate** pour ajouter la nouvelle partition en tant que PV.

```
[root@servera ~]# pvccreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

▶ **6.** Étendez le groupe de volumes.

- 6.1. Utilisez **vgextend** pour étendre le VG appelé `servera_01_vg`, en utilisant le nouveau PV `/dev/vdb3`.

```
[root@servera ~]# vgextend servera_01_vg /dev/vdb3
Volume group "servera_01_vg" successfully extended
```

- 6.2. Utilisez **vgdisplay** pour examiner à nouveau la place libre sur le VG `servera_01_vg`. L'espace disponible doit être suffisant à présent.

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name           servera_01_vg
System ID
Format           lvm2
...output omitted...
VG Size          1012.00 MiB
PE Size          4.00 MiB
```

**CHAPITRE 7 |** Gestion des volumes logiques

```
Total PE           253
Alloc PE / Size   100 / 400.00 MiB
Free  PE / Size   153 / 612.00 MiB
VG UUID          0BBATU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

612 Mio d'espace est maintenant disponible (153 PE x étendues 4 Mio).

- 7. Utilisez **lvextend** pour étendre le LV existant à 700 Mio.

```
[root@servera ~]# lvextend -L 700M /dev/servera_01_vg/servera_01_lv
  Size of logical volume servera_01_vg/servera_01_lv changed from 400.00 MiB (100
  extents) to 700.00 MiB (175 extents).
  Logical volume servera_01_vg/servera_01_lv successfully resized.
```

**NOTE**

L'exemple spécifie la taille exacte pour créer le LV final, mais vous pouvez aussi spécifier la quantité d'espace supplémentaire souhaitée :

- **-L +300M** pour ajouter le nouvel espace en indiquant la taille en Mio ;
- **-l 175** pour spécifier le nombre total d'étendues (175 PE x 4 Mio) ;
- **-l +75** pour ajouter les étendues supplémentaires nécessaires.

- 8. Utilisez **xfs\_growfs** pour étendre le système de fichiers XFS au reste de l'espace disponible sur le LV.

```
[root@servera ~]# xfs_growfs /data
meta-data=/dev/mapper/servera_01_vg-servera_01_lv isize=512    agcount=4,
  agsize=25600 blks
...output omitted...
```

- 9. Utilisez **df** et **ls | wc** pour vérifier la taille du nouveau système de fichiers et vous assurer que les fichiers qui existaient auparavant s'y trouvent encore.

```
[root@servera ~]# df -h /data
Filesystem              Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  695M   26M  670M   4% /data
[root@servera ~]# ls /data | wc -l
34
```

Les fichiers existent toujours et le système de fichiers se rapproche de la taille spécifiée.

- 10. Déconnectez-vous de servera.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur **workstation**, exéutez la commande **lab lvm-extending finish** pour mettre fin à l'exercice. Ce script supprime toutes les unités de stockage installées sur **servera** pendant l'exercice.

```
[student@workstation ~]$ lab lvm-extending finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# GESTION DES VOLUMES LOGIQUES

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans le cadre de cet atelier, vous allez redimensionner un volume logique existant, ajouter les ressources LVM nécessaires, puis ajouter un nouveau volume logique sur lequel un système de fichiers XFS est monté de façon persistante.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Redimensionner le volume logique `serverb_01_lv` sur 768 Mio.
- Créer un volume logique de 128 Mio appelé `serverb_02_lv`, avec un système de fichiers XFS, monté de façon persistante dans **/storage/data2**.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab lvm-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle prépare également l'espace de stockage sur `serverb` pour l'exercice.

```
[student@workstation ~]$ lab lvm-review start
```

Sur `serverb`, un volume logique appelé `serverb_01_lv` monté dans **/storage/data1** manque d'espace disque, et il vous a été demandé d'étendre sa taille à 768 Mio. Vous devez vous assurer que `serverb_01_lv` reste monté de manière persistante dans **/storage/data1**.

Vous avez également été invité à créer un volume logique de 128 Mio appelé `serverb_02_lv`, monté dans **/storage/data2**. Vous avez été chargé de formater le nouveau volume logique avec le système de fichiers XFS.

Le groupe de volumes `serverb_01_vg` contient les volumes logiques. Malheureusement, l'espace disponible est insuffisant pour étendre le volume logique existant et ajouter le nouveau. Une partition de 512 Mio a été créée précédemment sur `/dev/vdb`. Vous avez été invité à utiliser 512 Mio supplémentaires sur `/dev/vdb`. Vous devez créer la partition correspondante.

1. Créez une partition de 512 Mio sur `/dev/vdb`, initialisez-la en tant que volume physique, puis utilisez-la pour étendre le groupe de volumes `serverb_01_vg`.
2. Étendez le volume logique `serverb_01_lv` à 768 Mio, système de fichiers compris.
3. Dans le groupe de volumes existant, créez un volume logique appelé `serverb_02_lv` en lui attribuant une taille de 128 Mio. Ajoutez un système de fichiers XFS et montez-le de façon persistante dans **/storage/data2**.
4. Lorsque vous avez terminé, redémarrez votre machine `serverb`, puis exécutez la commande `lab lvm-review grade` depuis votre machine `workstation` pour vérifier votre travail.

Attendez la mise en route complète de `serverb`, puis procédez à l'évaluation.

## Évaluation

À partir de `workstation`, exécutez le script `lab lvm-review grade` pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab lvm-review grade
```

## Fin

Sur `workstation`, exécutez le script `lab lvm-review finish` pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab lvm-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# GESTION DES VOLUMES LOGIQUES

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans le cadre de cet atelier, vous allez redimensionner un volume logique existant, ajouter les ressources LVM nécessaires, puis ajouter un nouveau volume logique sur lequel un système de fichiers XFS est monté de façon persistante.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Redimensionner le volume logique `serverb_01_lv` sur 768 Mio.
- Créer un volume logique de 128 Mio appelé `serverb_02_lv`, avec un système de fichiers XFS, monté de façon persistante dans **/storage/data2**.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab lvm-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle prépare également l'espace de stockage sur `serverb` pour l'exercice.

```
[student@workstation ~]$ lab lvm-review start
```

Sur `serverb`, un volume logique appelé `serverb_01_lv` monté dans **/storage/data1** manque d'espace disque, et il vous a été demandé d'étendre sa taille à 768 Mio. Vous devez vous assurer que `serverb_01_lv` reste monté de manière persistante dans **/storage/data1**.

Vous avez également été invité à créer un volume logique de 128 Mio appelé `serverb_02_lv`, monté dans **/storage/data2**. Vous avez été chargé de formater le nouveau volume logique avec le système de fichiers XFS.

Le groupe de volumes `serverb_01_vg` contient les volumes logiques. Malheureusement, l'espace disponible est insuffisant pour étendre le volume logique existant et ajouter le nouveau. Une partition de 512 Mio a été créée précédemment sur **/dev/vdb**. Vous avez été invité à utiliser 512 Mio supplémentaires sur **/dev/vdb**. Vous devez créer la partition correspondante.

1. Créez une partition de 512 Mio sur **/dev/vdb**, initialisez-la en tant que volume physique, puis utilisez-la pour étendre le groupe de volumes `serverb_01_vg`.
  - 1.1. Connectez-vous à `serverb` en tant qu'utilisateur `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

**CHAPITRE 7 |** Gestion des volumes logiques

- 1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 1.3. Utilisez **parted** pour créer la partition de 512 Mio et lui attribuer le type « Linux LVM ».

```
[root@serverb ~]# parted -s /dev/vdb mkpart primary 514MiB 1026MiB  
[root@serverb ~]# parted -s /dev/vdb set 2 lvm on
```

- 1.4. Utilisez **udevadm settle** pour que le système enregistre la nouvelle partition.

```
[root@servera ~]# udevadm settle
```

- 1.5. Utilisez **pvcreate** pour initialiser la partition en tant que PV.

```
[root@serverb ~]# pvcreate /dev/vdb2  
Physical volume "/dev/vdb2" successfully created.
```

- 1.6. Utilisez **vgextend** pour étendre le VG appelé **serverb\_01\_vg**, en utilisant le nouveau PV **/dev/vdb2**.

```
[root@serverb ~]# vgextend serverb_01_vg /dev/vdb2  
Volume group "serverb_01_vg" successfully extended
```

2. Étendez le volume logique **serverb\_01\_lv** à 768 Mio, système de fichiers compris.

- 2.1. Utilisez **lvextend** pour étendre le LV **serverb\_01\_lv** à 768 Mio.

```
[root@serverb ~]# lvextend -L 768M /dev/serverb_01_vg/serverb_01_lv  
Size of logical volume serverb_01_vg/serverb_01_lv changed from 256.00 MiB (64  
extents) to 768.00 MiB (192 extents).  
Logical volume serverb_01_vg/serverb_01_lv successfully resized.
```

**NOTE**

Vous auriez également pu utiliser l'option **-L +512M** pour redimensionner le LV.

- 2.2. Utilisez **xfs\_growfs** pour étendre le système de fichiers XFS au reste de l'espace disponible sur le LV.

```
[root@serverb ~]# xfs_growfs /storage/data1  
meta-data=/dev/mapper/serverb_01_vg-serverb_01_lv isize=512    agcount=4,  
agsize=16384 blks  
...output omitted...
```

**NOTE**

Cet exemple présente l'utilisation de l'étape **xfs\_growfs** pour étendre le système de fichiers. Une autre solution consisterait à ajouter l'option **-r** à la commande **lvextend**.

3. Dans le groupe de volumes existant, créez un volume logique appelé **serverb\_02\_lv** en lui attribuant une taille de 128 Mio. Ajoutez un système de fichiers XFS et montez-le de façon persistante dans **/storage/data2**.
  - 3.1. Utilisez **lvcreate** pour créer un LV de 128 Mio appelé **serverb\_02\_lv** à partir du VG **serverb\_01\_vg**.

```
[root@serverb ~]# lvcreate -n serverb_02_lv -L 128M serverb_01_vg
Logical volume "serverb_02_lv" created
```

- 3.2. Utilisez **mkfs** pour placer un système de fichiers **xfs** sur le LV **serverb\_02\_lv**. Utilisez le nom de périphérique LV.

```
[root@serverb ~]# mkfs -t xfs /dev/serverb_01_vg/serverb_02_lv
meta-data=/dev/serverb_01_vg/serverb_02_lv isize=512    agcount=4, agsize=8192
blks
...output omitted...
```

- 3.3. Utilisez **mkdir** pour créer un point de montage dans **/storage/data2**.

```
[root@serverb ~]# mkdir /storage/data2
```

- 3.4. Ajoutez la ligne suivante à la fin de **/etc/fstab** sur **serverb**:

```
/dev/serverb_01_vg/serverb_02_lv  /storage/data2  xfs  defaults  1 2
```

- 3.5. Utilisez **systemctl daemon-reload** pour mettre à jour **systemd** avec la nouvelle configuration **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

- 3.6. Utilisez **mount** pour vérifier l'entrée **/etc/fstab** et monter le nouveau périphérique LV **serverb\_02\_lv**.

```
[root@serverb ~]# mount /storage/data2
```

4. Lorsque vous avez terminé, redémarrez votre machine **serverb**, puis exécutez la commande **lab lvm-review grade** depuis votre machine **workstation** pour vérifier votre travail.

```
[root@serverb ~]# systemctl reboot
```

Attendez la mise en route complète de **serverb**, puis procédez à l'évaluation.

## Évaluation

À partir de `workstation`, exéutez le script `lab lvm-review grade` pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab lvm-review grade
```

## Fin

Sur `workstation`, exéutez le script `lab lvm-review finish` pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab lvm-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- LVM vous permet de créer du stockage flexible en allouant de l'espace sur plusieurs périphériques de stockage.
- Les volumes physiques, les groupes de volumes et les volumes logiques sont gérés par divers outils tels que **pvccreate**, **vgreduce** et **lvextend**.
- Vous pouvez formater les volumes logiques avec un système de fichiers ou un espace d'échange et les monter de manière persistante.
- Du stockage supplémentaire peut être ajouté aux groupes de volumes et les volumes logiques peuvent être étendus de manière dynamique.



## CHAPITRE 8

# MISE EN ŒUVRE DE FONCTIONNALITÉS DE STOCKAGE AVANCÉES

### PROJET

Gérer le stockage à l'aide du système de gestion de stockage local Stratis et utiliser les volumes VDO pour optimiser l'espace de stockage utilisé.

### OBJECTIFS

- Gérer plusieurs couches de stockage à l'aide de la gestion de stockage local Stratis.
- Optimiser l'utilisation de l'espace de stockage en utilisant VDO pour compresser et dédupliquer les données sur les périphériques de stockage.

### SECTIONS

- Gestion du stockage en couches avec Stratis (et exercice guidé)
- Compression et déduplication du stockage avec VDO (et exercice guidé)

### ATELIER

Mise en œuvre de fonctionnalités de stockage avancées

# GESTION DU STOCKAGE EN COUCHES AVEC STRATIS

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir gérer plusieurs couches de stockage à l'aide de la gestion de stockage local Stratis.

## DESCRIPTION DE L'ARCHITECTURE STRATIS

La solution de stockage local actuelle dans Red Hat Enterprise Linux (RHEL) inclut de nombreuses technologies stables et matures, y compris le mappage de périphérique (dm), le gestionnaire de volumes logiques (LVM) et le système de fichiers XFS. Les fonctionnalités fournies par ces composants comprennent des systèmes de fichiers extrêmement évolutifs, des instantanés, des périphériques logiques (RAID) redondants, des chemins d'accès multiples, l'allocation fine et dynamique, la mise en mémoire cache, la déduplication et la prise en charge des machines virtuelles et des conteneurs. Chaque couche de pile de stockage (dm, LVM et XFS) est gérée à l'aide de commandes et d'utilitaires propres aux couches, obligeant les administrateurs système à gérer les périphériques physiques, les volumes à taille fixe et les systèmes de fichiers en tant que composants de stockage distincts.

Une nouvelle génération de solutions de gestion de stockage est apparue ces dernières années, appelée *systèmes de fichiers de gestion de volumes*, qui gèrent de manière dynamique et transparente la couche de volume lorsque des systèmes de fichiers sont créés et dimensionnés. Cependant, bien que le développement de ces systèmes de fichiers par la communauté ait duré des années, aucun n'a atteint le niveau de prise en charge et de stabilité fonctionnelles requis pour devenir le stockage local principal de Red Hat Enterprise Linux.

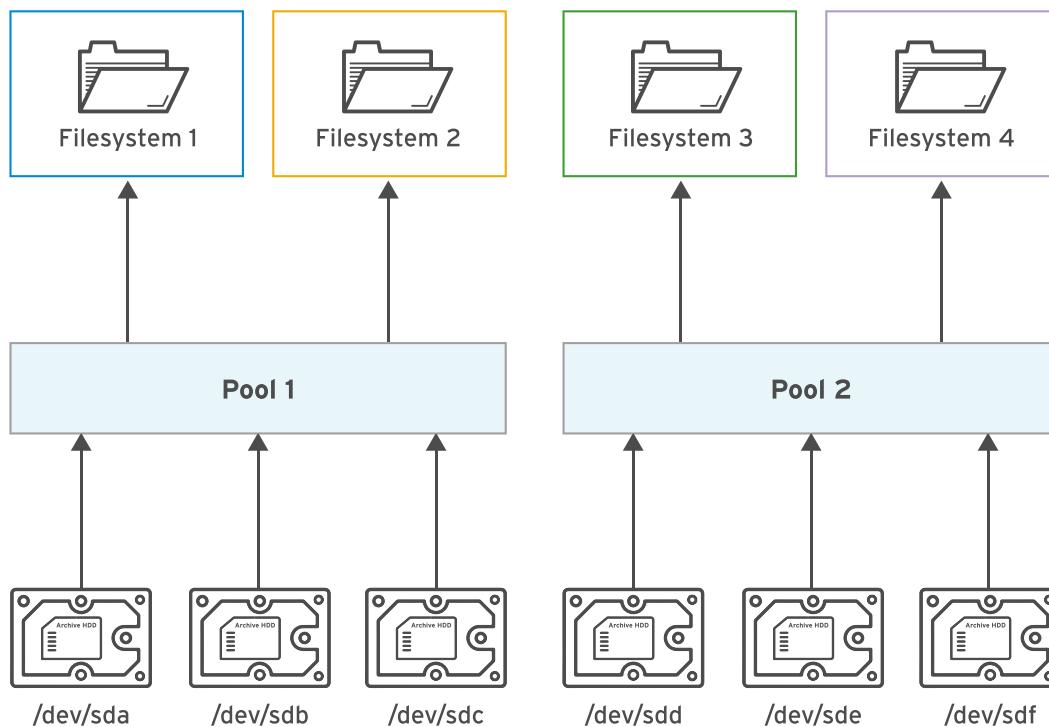
Avec RHEL 8, Red Hat présente la solution de gestion de stockage Stratis. Au lieu de développer à partir de zéro, comme l'ont essayé d'autres projets de stockage, Stratis utilise les composants de stockage RHEL existants. Stratis s'exécute en tant que service qui gère des pools de périphériques de stockage physiques, et crée et gère de manière transparente des volumes pour les systèmes de fichiers en cours de création. Étant donné que Stratis utilise les pilotes et les outils de stockage existants, toutes les fonctionnalités de stockage avancées que vous utilisez actuellement dans LVM, XFS et le mappage de périphérique sont également prises en charge par Stratis.

Dans un système de fichiers de gestion de volumes, les systèmes de fichiers sont construits à l'intérieur de *pools* partagés de périphériques de disques utilisant un concept connu sous le nom d'*allocation fine et dynamique*. Les systèmes de fichiers Stratis n'ont pas de taille fixe et ne préallouent plus l'espace de bloc non utilisé. Bien que le système de fichiers soit toujours construit sur un volume LVM caché, Stratis gère le volume sous-jacent pour vous et peut l'étendre en cas de besoin. La taille d'utilisation d'un système de fichiers est considérée comme la quantité de blocs réellement utilisée par les fichiers. L'espace disponible pour un système de fichiers est la quantité d'espace encore inutilisée dans les périphériques mis en pool sur lesquels il réside. Plusieurs systèmes de fichiers peuvent résider dans le même pool de périphériques de disque et partager l'espace disponible, mais les systèmes de fichiers peuvent également réserver de l'espace de pool pour garantir la disponibilité en cas de besoin.

Stratis utilise des métadonnées stockées pour reconnaître les pools, les volumes et les systèmes de fichiers gérés. Par conséquent, les systèmes de fichiers créés par Stratis ne doivent jamais être reformatés ou reconfigurés manuellement ; ils ne doivent être gérés qu'à l'aide d'outils et de commandes Stratis. La configuration manuelle des systèmes de fichiers Stratis peut entraîner

la perte de ces métadonnées et empêcher Stratis de reconnaître les systèmes de fichiers qu'il a créés.

Vous pouvez créer plusieurs pools avec différents ensembles de périphériques en mode bloc. À partir de chaque pool, vous pouvez créer un ou plusieurs systèmes de fichiers. Actuellement, vous pouvez créer jusqu'à  $2^{24}$  systèmes de fichiers par pool. Le diagramme suivant illustre le positionnement des éléments de la solution de gestion de stockage Stratis.



**Figure 8.1: Éléments de Stratis**

Un pool regroupe des périphériques en mode bloc dans le *niveau de données* et éventuellement le *niveau de cache*. Le niveau de données met l'accent sur la flexibilité et l'intégrité, et le niveau de cache sur l'amélioration des performances. Le niveau de cache étant destiné à améliorer les performances, vous devez utiliser des périphériques en mode bloc qui comportent des IOPS (opérations d'entrée/sortie par seconde) élevées, tels que les SSD.

## Description de la pile de stockage simplifiée

Stratis simplifie de nombreux aspects de la configuration et de l'allocation de stockage local des produits Red Hat. Par exemple, dans les versions antérieures du programme d'installation Anaconda, les administrateurs système devaient superposer chaque aspect de la gestion de disque. À présent, le programme d'installation utilise Stratis, ce qui simplifie la configuration du disque. Les autres produits qui utilisent Stratis incluent Cockpit, Red Hat Virtualization et Red Hat Enterprise Linux Atomic Host. Pour tous ces produits, Stratis simplifie la gestion de l'espace de stockage et des instantanés, en évitant les erreurs. Stratis permet une intégration plus facile avec les outils de gestion de plus haut niveau qu'en utilisant une interface en ligne de commande par programmation.

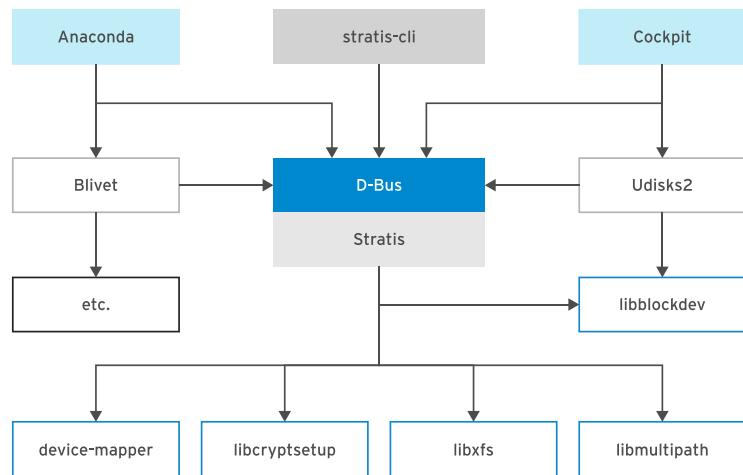


Figure 8.2: Stratis dans la pile de gestion du stockage Linux

## Description des couches Stratis

En interne, Stratis utilise le sous-système **Backstore** pour gérer les périphériques en mode bloc et le sous-système **Thinpool** pour gérer les pools. Le sous-système **Backstore** comprend un niveau de données qui conserve les métadonnées sur disque sur des périphériques en mode bloc, et qui détecte et corrige la corruption de données. Le niveau de cache utilise des périphériques en mode bloc à hautes performances pour faire office de cache au-dessus du niveau de données. Le sous-système **Thinpool** gère les volumes alloués de manière fine et dynamique associés aux systèmes de fichiers Stratis. Ce sous-système utilise le pilote de mappeur de périphérique **dm-thin** pour remplacer LVM lors du dimensionnement et de la gestion de volumes virtuels. **dm-thin** crée des volumes de grande taille virtuelle, formatés avec XFS, mais de petite taille physique. Lorsque la limite supérieure de la taille physique est presque atteinte, Stratis l'élargit automatiquement.

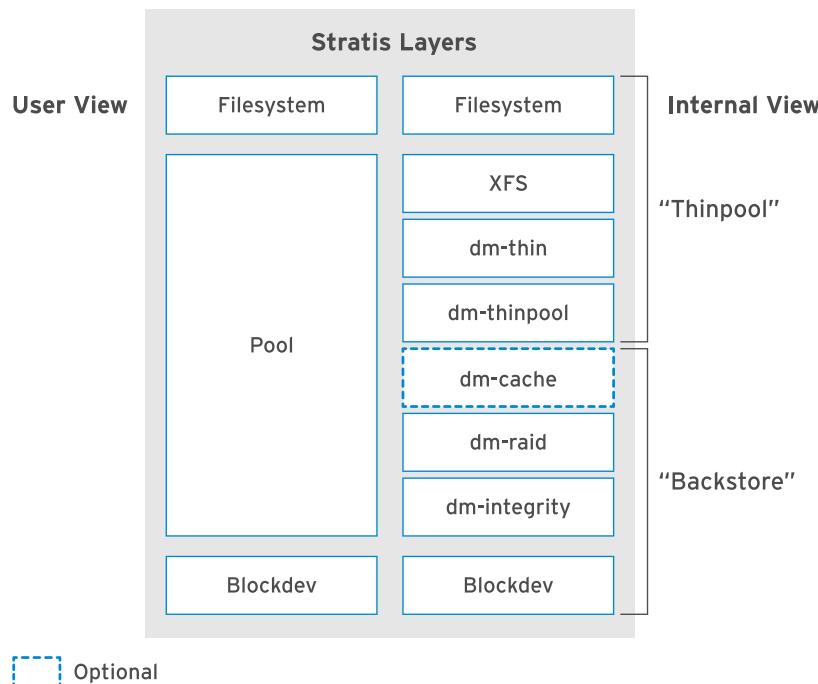


Figure 8.3: Couches Stratis

## Gestion des systèmes de fichiers alloués de manière fine et dynamique

Pour gérer les systèmes de fichiers alloués de manière fine et dynamique en utilisant la solution de gestion de stockage Stratis, installez les paquetages *stratis-cli* et *stratisd*. Le paquetage *stratis-cli* fournit la commande **stratis**, qui traduit les demandes de l'utilisateur au service *stratisd* via l'API D-Bus. Le paquetage *stratisd* fournit le service *stratisd* qui met en œuvre l'interface D-Bus, et gère et surveille les éléments de Stratis, tels que les périphériques en mode bloc, les pools et les systèmes de fichiers. L'API D-Bus est disponible si le service *stratisd* est en cours d'exécution.

Installez et activez Stratis en utilisant les outils habituels :

- Installez les paquetages *stratisd* et *stratis-cli* à l'aide de la commande **yum install**.

```
[root@host ~]# yum install stratis-cli stratisd
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- Activez le service *stratisd* à l'aide de la commande **systemctl**.

```
[root@host ~]# systemctl enable --now stratisd
```

Les opérations de gestion courantes effectuées à l'aide de la solution de gestion de stockage Stratis sont les suivantes.

- Créez des pools composés d'un ou de plusieurs périphériques en mode bloc à l'aide de la commande **stratis pool create**.

```
[root@host ~]# stratis pool create pool1 /dev/vdb
```

Chaque pool est un sous-répertoire sous le répertoire **/stratis**.

- Utilisez la commande **stratis pool list** pour afficher la liste des pools disponibles.

```
[root@host ~]# stratis pool list
Name      Total Physical Size  Total Physical Used
pool1          5 GiB            52 MiB
```

- Utilisez la commande **stratis pool add-data** pour ajouter des périphériques en mode bloc à un pool.

```
[root@host ~]# stratis pool add-data pool1 /dev/vdc
```

- Utilisez la commande **stratis blockdev list** pour afficher les périphériques en mode bloc d'un pool.

```
[root@host ~]# stratis blockdev list pool1
Pool Name  Device Node      Physical Size   State  Tier
pool1      /dev/vdb           5 GiB        In-use  Data
pool1      /dev/vdc           5 GiB        In-use  Data
```

- Utilisez la commande **stratis filesystem create** pour créer un système de fichiers dynamique et flexible à partir d'un pool.

```
[root@host ~]# stratis filesystem create pool1 filesystem1
```

Les liens vers les systèmes de fichiers Stratis se trouvent dans le répertoire **/stratis/pool1**.

- Stratis prend en charge la création d'instantanés du système de fichiers à l'aide de la commande **stratis filesystem snapshot**. Les instantanés sont indépendants des systèmes de fichiers sources.

```
[root@host ~]# stratis filesystem snapshot pool1 filesystem1 snapshot1
```

- Utilisez la commande **stratis filesystem list** pour afficher la liste des systèmes de fichiers disponibles.

```
[root@host ~]# stratis filesystem list
...output omitted...
```

Pour vous assurer que les systèmes de fichiers Stratis sont montés de manière persistante, éditez **/etc/fstab** et spécifiez les détails du système de fichiers. La commande suivante affiche l'UUID du système de fichiers que vous devez utiliser dans **/etc/fstab** pour identifier le système de fichiers.

```
[root@host ~]# lsblk --output=UUID /stratis/pool1/filesystem1
UUID
31b9363b-add8-4b46-a4bf-c199cd478c55
```

L'exemple suivant est une entrée dans le fichier **/etc/fstab** destinée à monter un système de fichiers Stratis de manière persistante.

```
UUID=31b9...8c55 /dir1 xfs defaults,x-systemd.requires=stratisd.service 0 0
```

L'option de montage **x-systemd.requires=stratisd.service** retarde le montage du système de fichiers jusqu'à ce que systemd démarre stratisd.service pendant le processus d'amorçage.



### NOTE

Si vous n'utilisez pas l'option de montage **x-systemd.requires=stratisd.service** dans **/etc/fstab** pour le système de fichiers Stratis, cela entraînera le lancement de la machine vers **emergency.target** au prochain redémarrage.



### RÉFÉRENCES

Pour plus d'informations, reportez-vous au chapitre *Managing layered local storage with Stratis* du guide *Red Hat Enterprise Linux 8 Configuring and Managing File Systems* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_and\\_managing\\_file\\_systems/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_file_systems/)

#### Stockage Stratis

<https://stratis-storage.github.io/>

#### Enseignements de ZFS, Btrfs et de Linux Volume Manager

<https://opensource.com/article/18/4/stratis-lessons-learned>

## ► EXERCICE GUIDÉ

# GESTION DU STOCKAGE EN COUCHES AVEC STRATIS

Au cours de cet exercice, vous utiliserez la solution de gestion de stockage Stratis pour créer des pools, des volumes et des systèmes de fichiers fonctionnant en coopération.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un système de fichiers alloué de manière fine et dynamique à l'aide de la solution de gestion de stockage Stratis.
- Vérifier que les volumes Stratis augmentent de manière dynamique pour prendre en charge l'accroissement des données en temps réel.
- Accéder aux données à partir de l'instantané d'un système de fichiers alloué de manière fine et dynamique.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant que `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab advstorage-stratis start` pour démarrer l'exercice. Ce script configure correctement l'environnement et garantit que les disques supplémentaires sur `servera` ont de l'espace disponible.

```
[student@workstation ~]$ lab advstorage-stratis start
```

- 1. À partir de `workstation`, ouvrez une session SSH sur `servera` en tant que `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Basculez vers l'utilisateur `root`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Installez les paquetages `stratisd` et `stratis-cli` à l'aide de `yum`.

```
[root@servera ~]# yum install stratisd stratis-cli
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- ▶ 4. Activez le service **stratisd** à l'aide de la commande **systemctl**.

```
[root@servera ~]# systemctl enable --now stratisd
```

- ▶ 5. Assurez-vous que le pool Stratis **stratispool1** existe avec le périphérique en mode bloc **/dev/vdb**.

- 5.1. Créez un pool Stratis nommé **stratispool1** en utilisant la commande **stratis pool create**.

```
[root@servera ~]# stratis pool create stratispool1 /dev/vdb
```

- 5.2. Vérifiez la disponibilité de **stratispool1** en utilisant la commande **stratis pool list**.

```
[root@servera ~]# stratis pool list
Name          Total Physical Size  Total Physical Used
stratispool1           5 GiB            52 MiB
```

Remarquez la taille du pool dans la sortie précédente.

- ▶ 6. Augmentez la capacité de **stratispool1** en utilisant le périphérique en mode bloc **/dev/vdc**.

- 6.1. Ajoutez le périphérique en mode bloc **/dev/vdc** à **stratispool1** à l'aide de la commande **stratis pool add-data**.

```
[root@servera ~]# stratis pool add-data stratispool1 /dev/vdc
```

- 6.2. Vérifiez la taille de **stratispool1** en utilisant la commande **stratis pool list**.

```
[root@servera ~]# stratis pool list
Name          Total Physical Size  Total Physical Used
stratispool1           10 GiB           56 MiB
```

Comme indiqué ci-dessus, la taille du pool **stratispool1** a augmenté lorsque vous avez ajouté le périphérique en mode bloc.

- 6.3. Vérifiez les périphériques en mode bloc qui sont actuellement membres de **stratispool1** en utilisant la commande **stratis blockdev list**.

```
[root@servera ~]# stratis blockdev list stratispool1
Pool Name      Device Node    Physical Size   State   Tier
stratispool1   /dev/vdb          5 GiB   In-use   Data
stratispool1   /dev/vdc          5 GiB   In-use   Data
```

- ▶ 7. Ajoutez un système de fichiers alloué de manière fine et dynamique nommé **stratis-filesystem1** dans le pool **stratispool1**. Montez le système de fichiers sur **/stratisvol**. Créez un fichier dans le système de fichiers **stratis-filesystem1** nommé **file1** contenant le texte **Hello World!**.

- 7.1. Créez le système de fichiers alloué de manière fine et dynamique **stratis-filesystem1** dans **stratispool1** en utilisant la commande **stratis filesystem create**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@servera ~]# stratis filesystem create stratispool1 stratis-filesystem1
```

- 7.2. Vérifiez la disponibilité de **stratis-filesystem1** en utilisant la commande **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name           Used     Created        Device
                  UUID
stratispool1   stratis-filesystem1 546 MiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-filesystem1 8714...e7db
```

Remarquez l'utilisation actuelle de **stratis-filesystem1**. Cette utilisation du système de fichiers augmente à la demande dans les étapes suivantes.

- 7.3. Créez un répertoire nommé **/stratisvol** en utilisant la commande **mkdir**.

```
[root@servera ~]# mkdir /stratisvol
```

- 7.4. Montez **stratis-filesystem1** sur **/stratisvol** en utilisant la commande **mount**.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-filesystem1 /stratisvol
```

- 7.5. Vérifiez que le système de fichiers Stratis **stratis-filesystem1** est monté sur **/stratisvol** en utilisant la commande **mount**.

```
[root@servera ~]# mount
...output omitted...
/dev/mapper/stratis-1-5c0e...12b9-thin-fs-8714...e7db on /stratisvol type xfs
(rw,relatime,seclabel,attr2,inode64,sunit=2048,swidth=2048,noquota)
```

- 7.6. Créez le fichier texte **/stratisvol/file1** en utilisant la commande **echo**.

```
[root@servera ~]# echo "Hello World!" > /stratisvol/file1
```

- 8. Vérifiez que le système de fichiers **stratis-filesystem1** alloué de manière fine et dynamique croît réellement de manière dynamique à mesure que les données sur le système de fichiers augmentent.

- 8.1. Consultez l'utilisation actuelle de **stratis-filesystem1** en utilisant la commande **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
          UUID
stratispool1  stratis-filesystem1  546 MiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-filesystem1  8714...e7db
```

- 8.2. Créez un fichier de 2 Gio sur **stratis-filesystem1** en utilisant la commande **dd**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@servera ~]# dd if=/dev/urandom of=/stratisvol/file2 bs=1M count=2048
```

- 8.3. Vérifiez l'utilisation de **stratis-filesystem1** à l'aide de la commande **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
          UUID
stratispool1  stratis-filesystem1  2.53 GiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-filesystem1  8714...e7db
```

La sortie précédente montre que l'utilisation de **stratis-filesystem1** a augmenté. L'augmentation de l'utilisation confirme que le système de fichiers alloué de manière fine et dynamique s'est développé de manière dynamique pour répondre aux besoins de croissance des données en temps réel que vous avez entraînés en créant **/stratisvol/file2**.

- 9. Créez un instantané de **stratis-filesystem1** nommé **stratis-filesystem1-snap**. L'instantané vous donne accès à tout fichier supprimé de **stratis-filesystem1**.

- 9.1. Créez un instantané de **stratis-filesystem1** en utilisant la commande **stratis filesystem snapshot**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@servera ~]# stratis filesystem snapshot stratispool1 \
stratis-filesystem1 stratis-filesystem1-snap
```

- 9.2. Vérifiez la disponibilité de l'instantané en utilisant la commande **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
...output omitted...
stratispool1  stratis-filesystem1-snap  2.53 GiB  Mar 29 2019 10:28  /stratis/
stratispool1/stratis-filesystem1-snap  291d...8a16
```

- 9.3. Supprimez le fichier **/stratisvol/file1**.

```
[root@servera ~]# rm /stratisvol/file1  
rm: remove regular file '/stratisvol/file1'? y
```

9.4. Créez le répertoire **/stratisvol-snap** en utilisant la commande **mkdir**.

```
[root@servera ~]# mkdir /stratisvol-snap
```

9.5. Montez l'instantané **stratis-filesystem1-snap** sur **/stratisvol-snap** en utilisant la commande **mount**.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-filesystem1-snap \  
/stratisvol-snap
```

9.6. Vérifiez que vous pouvez toujours accéder au fichier que vous avez supprimé de **stratis-filesystem1** en utilisant l'instantané **stratis-filesystem1-snap**.

```
[root@servera ~]# cat /stratisvol-snap/file1  
Hello World!
```

► 10. Démontez **/stratisvol** et **/stratisvol-snap** en utilisant la commande **umount**.

```
[root@servera ~]# umount /stratisvol-snap  
[root@servera ~]# umount /stratisvol
```

► 11. Supprimez le système de fichiers alloué de manière fine et dynamique **stratis-filesystem1** et son instantané **stratis-filesystem1-snap** du système.

11.1. Supprimez définitivement l'instantané **stratis-filesystem1-snap** en utilisant la commande **stratis filesystem destroy**.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1-snap
```

11.2. Supprimez définitivement **stratis-filesystem1** en utilisant la commande **stratis filesystem destroy**.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1
```

11.3. Quittez le shell root de l'utilisateur.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

11.4. Déconnectez-vous de **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez **lab advstorage-stratis finish** pour mettre fin à l'exercice. Ce script supprime les partitions et les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab advstorage-stratis finish
```

L'exercice guidé est maintenant terminé.

# COMPRESSION ET DÉDUPLICATION DU STOCKAGE AVEC VDO

---

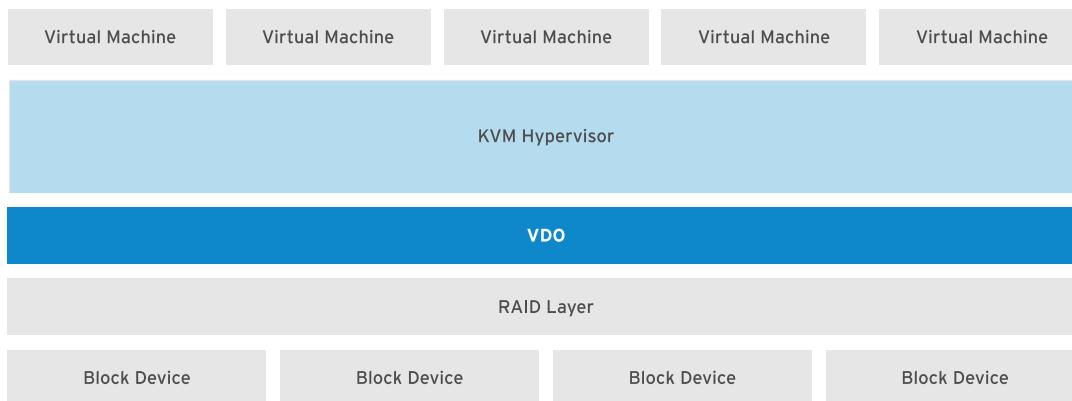
## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir optimiser l'utilisation de l'espace de stockage en utilisant VDO pour compresser et dédupliquer les données sur les périphériques de stockage.

## DESCRIPTION DE VIRTUAL DATA OPTIMIZER

Red Hat Enterprise Linux 8 comprend le pilote VDO (Virtual Data Optimizer) qui optimise l'empreinte des données sur les périphériques en mode bloc. VDO est un pilote de mappeur de périphérique Linux qui réduit l'utilisation de l'espace disque sur les périphériques en mode bloc et minimise la réplication des données, en économisant de l'espace disque et en augmentant même le débit de données. VDO comprend deux modules de noyau : le module `kvdo` pour contrôler de manière transparente la compression des données et le module `uds` affecté à la déduplication.

La couche VDO est placée sur un périphérique de stockage en mode bloc existant, tel qu'un périphérique RAID ou un disque local. Ces périphériques en mode bloc peuvent également être des périphériques chiffrés. Les couches de stockage, telles que les volumes logiques LVM et les systèmes de fichiers, sont placées sur un périphérique VDO. Le diagramme suivant illustre le positionnement de VDO dans une infrastructure composée de machines virtuelles KVM utilisant des périphériques de stockage optimisés.



**Figure 8.4: Machines virtuelles basées sur VDO**

VDO applique trois phases aux données dans l'ordre suivant pour réduire l'empreinte sur les périphériques de stockage :

1. L'*élimination des blocs composés de zéros* filtre les blocs de données qui ne contiennent que des zéros (0) et enregistre les informations de ces blocs uniquement dans les métadonnées. Les blocs de données qui ne sont pas composés de zéros sont ensuite transmis à la phase suivante du traitement. Cette phase active la fonctionnalité d'allocation fine et dynamique dans les périphériques VDO.
2. La *déduplication* élimine les blocs de données redondants. Lorsque vous créez plusieurs copies des mêmes données, VDO détecte les blocs de données en double et met à jour les métadonnées pour utiliser ces blocs en tant que références au bloc de données d'origine sans

créer de blocs de données redondants. Le module de noyau UDS (Universal Deduplication Service) vérifie la redondance des données via les métadonnées qu'il conserve. Ce module de noyau est fourni avec le VDO.

3. La *compression* est la dernière phase. Le module de noyau `kvdo` compresse les blocs de données à l'aide de la compression LZ4 et les regroupe sur des blocs de 4 Ko.

## MISE EN ŒUVRE DE VIRTUAL DATA OPTIMIZER

Les périphériques logiques que vous créez avec VDO sont appelés *Volumes VDO*. Les volumes VDO sont similaires aux partitions de disque. Vous pouvez formater les volumes avec le type de système de fichiers souhaité et les monter comme un système de fichiers classique. Vous pouvez également utiliser un volume VDO en tant que volume physique LVM.

Pour créer un volume VDO, spécifiez un périphérique en mode bloc et le nom du périphérique logique que VDO présente à l'utilisateur. Vous pouvez éventuellement spécifier la taille logique du volume VDO. La taille logique du volume VDO peut être supérieure à la taille physique du périphérique en mode bloc réel.

Étant donné que les volumes VDO sont alloués de manière fine et dynamique, les utilisateurs ne peuvent voir que l'espace logique utilisé et ne sont pas conscients de l'espace physique réel disponible. Si vous ne spécifiez pas la taille logique lors de la création du volume, VDO considère que la taille physique réelle est la taille logique du volume. Ce ratio 1:1 entre la taille logique et la taille physique de la mise en correspondance donne de meilleures performances mais offre une utilisation moins efficace de l'espace de stockage. En fonction des besoins de votre infrastructure, vous devez définir des priorités en termes de performances ou d'utilisation optimale de l'espace.

Lorsque la taille logique d'un volume VDO est supérieure à la taille physique réelle, vous devez surveiller de manière proactive les statistiques de volume pour afficher l'utilisation réelle à l'aide de la commande `vdostats --verbose`.

### Activation de VDO

Installez les paquetages `vdo` et `kmod-kvdo` pour activer VDO dans le système.

```
[root@host ~]# yum install vdo kmod-kvdo
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

### Création d'un volume VDO

Pour créer un volume VDO, exécutez la commande `vdo create`.

```
[root@host ~]# vdo create --name=vdo1 --device=/dev/vdd --vdoLogicalSize=50G
...output omitted...
```

Si vous omettez la taille logique, le volume VDO résultant aura la même taille que son périphérique physique.

Lorsque le volume VDO est en place, vous pouvez le formater avec le type de système de fichiers de votre choix et le monter sous la hiérarchie du système de fichiers sur votre système.

## Analyse d'un volume VDO

Pour analyser un volume VDO, exécutez la commande **vdo status**. Cette commande affiche un rapport sur le système VDO et l'état du volume du VDO au format YAML. Il affiche également les attributs du volume VDO. Utilisez l'option **--name=** pour spécifier le nom d'un volume particulier. Si vous omettez le nom du volume spécifique, la sortie de la commande **vdo status** affiche l'état de tous les volumes VDO.

```
[root@host ~]# vdo status --name=vdo1  
...output omitted...
```

La commande **vdo list** affiche la liste des volumes VDO qui sont démarrés. Vous pouvez démarrer et arrêter un volume VDO à l'aide des commandes **vdo start** et **vdo stop**, respectivement.



### RÉFÉRENCES

Pour plus d'informations, consultez le chapitre *Getting started with VDO* du guide *Red Hat Enterprise Linux 8 Deduplicating and Compressing Storage* à l'adresse [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/deduplicating\\_and\\_compressing\\_storage/](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/deduplicating_and_compressing_storage/)

### Présentation de Virtual Data Optimizer

<https://rhelblog.redhat.com/2018/04/11/introducing-virtual-data-optimizer-to-reduce-cloud-and-on-premise-storage-costs/>

## ► EXERCICE GUIDÉ

# COMPRESSION ET DÉDUPLICATION DU STOCKAGE AVEC VDO

Dans cet exercice, vous allez créer un volume VDO, le formater avec un système de fichiers, le monter, y stocker des données et étudier l'impact de la compression et de la déduplication sur l'espace de stockage réellement utilisé.

## RÉSULTATS

Vous serez en mesure de réaliser les tâches suivantes :

- Créer un volume à l'aide de Virtual Data Optimizer, le formater avec un type de système de fichiers et y monter un système de fichiers.
- Examiner l'impact de la déduplication et de la compression des données sur un volume Virtual Data Optimizer.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant que `student` avec le mot de passe `student`.

Sur `workstation`, exécutez **`lab advstorage-vdo start`** pour démarrer l'exercice. Ce script permet de s'assurer qu'il n'existe aucune partition sur le disque `/dev/vdd` et configure correctement l'environnement.

```
[student@workstation ~]$ lab advstorage-vdo start
```

- 1. À partir de `workstation`, ouvrez une session SSH sur `servera` en tant que `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Créez le volume VDO `vdo1` en utilisant le périphérique `/dev/vdd`. Définissez sa taille logique sur 50 Go.

2.1. Basculez vers l'utilisateur `root`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2.2. Vérifiez que le paquetage `vdo` est installé en utilisant la commande `rpm`.

```
[root@servera ~]# yum list installed vdo
vdo-6.2.0.293-10.el8.x86_64
```

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

2.3. Créez le volume vdo1 à l'aide de la commande **vdo create**.

```
[root@servera ~]# vdo create --name=vdo1 --device=/dev/vdd --vdoLogicalSize=50G  
...output omitted...
```

2.4. Vérifiez la disponibilité du volume vdo1 en utilisant la commande **vdo list**.

```
[root@servera ~]# vdo list  
vdo1
```

► 3. Vérifiez que les fonctions de compression et de déduplication sont activées sur le volume vdo1.

3.1. Utilisez **grep** pour rechercher les lignes contenant la chaîne **Déduplication** dans la sortie de la commande **vdo status --name=vdo1**.

```
[root@servera ~]# vdo status --name=vdo1 | grep Deduplication  
Deduplication: enabled
```

3.2. Utilisez **grep** pour rechercher les lignes contenant la chaîne **Compression** dans la sortie de la commande **vdo status --name=vdo1**.

```
[root@servera ~]# vdo status --name=vdo1 | grep Compression  
Compression: enabled
```

► 4. Formatez le volume vdo1 avec le type de système de fichiers XFS et montez-le sur **/mnt/vdo1**.

4.1. Formatez le volume vdo1 avec le système de fichiers XFS en utilisant la commande **mkfs**.

```
[root@servera ~]# mkfs.xfs -K /dev/mapper/vdo1  
...output omitted...
```

L'option **-K** dans la précédente commande **mkfs.xfs** empêche les blocs inutilisés du système de fichiers d'être immédiatement rejetés, ce qui permet à la commande de revenir plus rapidement.

4.2. Utilisez la commande **udevadm** pour enregistrer le nouveau nœud de périphérique.

```
[root@servera ~]# udevadm settle
```

4.3. Créez le répertoire **/mnt/vdo1** en utilisant la commande **mkdir**.

```
[root@servera ~]# mkdir /mnt/vdo1
```

4.4. Montez le volume vdo1 sur **/mnt/vdo1** à l'aide de la commande **mount**.

```
[root@servera ~]# mount /dev/mapper/vdo1 /mnt/vdo1
```

4.5. Vérifiez que le volume vdo1 est correctement monté à l'aide de la commande **mount**.

```
[root@servera ~]# mount
...output omitted...
/dev/mapper/vdo1 on /mnt/vdo1 type xfs
(rw,relatime,seclabel,attr2,inode64,noquota)
```

- ▶ 5. Créez trois copies du fichier nommé **/root/install.img** sur le volume vdo1. Comparez les statistiques du volume pour vérifier la déduplication et la compression des données sur le volume. La sortie précédente peut différer sur votre système.
- 5.1. Affichez les statistiques initiales et l'état du volume à l'aide de la commande **vdostats**.

```
[root@servera ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/vdo1   5.0G    3.0G      2.0G  60%      99%
```

Notez que 3 Go du volume sont déjà utilisés, car lors de sa création, le volume VDO se réserve 3 à 4 Go pour lui-même. Notez également que la valeur **99%** du champ **Space saving%** indique que vous n'avez créé jusqu'à présent aucun contenu dans le volume, ce qui vous permet de disposer de tout l'espace de volume enregistré.

- 5.2. Copiez **/root/install.img** dans **/mnt/vdo1/install.img.1** et vérifiez les statistiques du volume. La copie du fichier peut prendre jusqu'à une minute.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.1
[root@servera ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/vdo1   5.0G    3.4G      1.6G  68%      5%
```

Notez que la valeur du champ **Used** est passée de **3.0G** à **3.4G**, car vous avez copié un fichier dans le volume et que cela occupe de l'espace. Notez également que la valeur du champ **Space saving%** est passée de **99%** à **5%**. En effet, le volume d'origine ne comportait aucun contenu, et cette valeur reflétait la faible utilisation de l'espace du volume et la disponibilité élevée de l'espace jusqu'à la création d'un fichier. La disponibilité de l'espace du volume est assez faible, car vous avez créé une copie unique du fichier dans le volume, et l'espace restant ne permet pas la déduplication.

- 5.3. Copiez **/root/install.img** dans **/mnt/vdo1/install.img.2** et vérifiez les statistiques du volume. La copie du fichier peut prendre jusqu'à une minute.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.2
[root@servera ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/vdo1   5.0G    3.4G      1.6G  68%      51%
```

Notez que l'espace du volume utilisé n'a pas été modifié, mais plutôt le pourcentage de l'espace de volume enregistré qui a augmenté, prouvant ainsi que la déduplication des données s'est produite afin de réduire la consommation d'espace pour les copies redondantes du même fichier. La valeur de **Space saving%** dans la sortie précédente peut différer sur votre système.

- 5.4. Quittez le shell root de l'utilisateur.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

5.5. Déconnectez-vous de servera.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Fin

Sur workstation, exéutez **lab advstorage-vdo finish** pour mettre fin à l'exercice. Ce script supprime les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab advstorage-vdo finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# MISE EN ŒUVRE DE FONCTIONNALITÉS DE STOCKAGE AVANCÉES

Dans cet exercice, vous allez utiliser la solution de gestion de stockage Stratis pour créer des systèmes de fichiers évolutifs afin de répondre aux demandes croissantes en données, et Virtual Data Optimizer pour créer des volumes permettant une utilisation efficace de l'espace de stockage.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un système de fichiers alloué de manière fine et dynamique à l'aide de la solution de gestion de stockage Stratis.
- Vérifier que les volumes Stratis augmentent de manière dynamique pour prendre en charge l'accroissement des données en temps réel.
- Accéder aux données à partir de l'instantané d'un système de fichiers alloué de manière fine et dynamique.
- Crée un volume à l'aide de Virtual Data Optimizer et le monter sur un système de fichiers.
- Examiner l'impact de la déduplication et de la compression des données sur un volume Virtual Data Optimizer.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant que `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab advstorage-review start` pour commencer l'atelier. Ce script configure correctement l'environnement et garantit que les disques supplémentaires sur `serverb` ont de l'espace disponible.

```
[student@workstation ~]$ lab advstorage-review start
```

1. À partir de `workstation`, ouvrez une session SSH sur `serverb` en tant que `student`.
2. Basculez vers l'utilisateur `root`.
3. Installez les paquetages `stratisd` et `stratis-cli` à l'aide de `yum`.
4. Démarrez et activez le service `stratisd` à l'aide de la commande `systemctl`.
5. Créez le pool Stratis `labpool` contenant le périphérique en mode bloc `/dev/vdb`.
6. Augmentez la capacité de `labpool` en utilisant le disque `/dev/vdc` disponible dans le système.
7. Créez un système de fichiers nommé `labfs` alloué de manière fine et dynamique dans le pool `labpool`. Montez ce système de fichiers sur `/labstratisvol` afin qu'il persiste lors des redémarrages. Créez un fichier nommé `labfile1` contenant le texte `Hello World!`

## CHAPITRE 8 | Mise en œuvre de fonctionnalités de stockage avancées

sur le système de fichiers `labfs`. N'oubliez pas d'utiliser l'option de montage `x-systemd.requires=stratisd.service` dans `/etc/fstab`.

8. Vérifiez que le système de fichiers `labfs` alloué de manière fine et dynamique croît réellement de manière dynamique à mesure que les données sur le système de fichiers augmentent.
9. Créez un instantané nommé `labfs-snap` du système de fichiers `labfs`. L'instantané vous permet d'accéder à n'importe quel fichier supprimé de `labfs`.
10. Créez le volume VDO `labvdo`, avec `/dev/vdd` comme périphérique. Définissez sa taille logique sur **50 Go**.
11. Montez le volume `labvdo` sur **/labvdovol1** avec le système de fichiers XFS afin qu'il persiste lors des redémarrages. N'oubliez pas d'utiliser l'option de montage `x-systemd.requires=stratisd.service` dans `/etc/fstab`.
12. Créez trois copies du fichier nommé **/root/install.img** sur le volume `labvdo`. Comparez les statistiques du volume pour vérifier la déduplication et la compression des données sur le volume.

## Évaluation

Sur `workstation`, exécutez la commande `lab advstorage-review grade` pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab advstorage-review grade
```

## Terminer

Sur `workstation`, exécutez `lab advstorage-review finish` pour mettre fin à l'exercice. Ce script supprime les partitions et les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab advstorage-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# MISE EN ŒUVRE DE FONCTIONNALITÉS DE STOCKAGE AVANCÉES

Dans cet exercice, vous allez utiliser la solution de gestion de stockage Stratis pour créer des systèmes de fichiers évolutifs afin de répondre aux demandes croissantes en données, et Virtual Data Optimizer pour créer des volumes permettant une utilisation efficace de l'espace de stockage.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un système de fichiers alloué de manière fine et dynamique à l'aide de la solution de gestion de stockage Stratis.
- Vérifier que les volumes Stratis augmentent de manière dynamique pour prendre en charge l'accroissement des données en temps réel.
- Accéder aux données à partir de l'instantané d'un système de fichiers alloué de manière fine et dynamique.
- Créer un volume à l'aide de Virtual Data Optimizer et le monter sur un système de fichiers.
- Examiner l'impact de la déduplication et de la compression des données sur un volume Virtual Data Optimizer.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez **lab advstorage-review start** pour commencer l'atelier. Ce script configure correctement l'environnement et garantit que les disques supplémentaires sur **serverb** ont de l'espace disponible.

```
[student@workstation ~]$ lab advstorage-review start
```

1. À partir de **workstation**, ouvrez une session SSH sur **serverb** en tant que **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

2. Basculez vers l'utilisateur **root**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. Installez les paquetages **stratisd** et **stratis-cli** à l'aide de **yum**.

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

```
[root@serverb ~]# yum install stratisd stratis-cli  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

4. Démarrez et activez le service stratisd à l'aide de la commande **systemctl**.

```
[root@serverb ~]# systemctl enable --now stratisd
```

5. Créez le pool Stratis labpool contenant le périphérique en mode bloc /dev/vdb.

- 5.1. Créez le pool Stratis labpool à l'aide de la commande **stratis pool create**.

```
[root@serverb ~]# stratis pool create labpool /dev/vdb
```

- 5.2. Vérifiez la disponibilité de labpool en utilisant la commande **stratis pool list**.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
labpool        5 GiB                 52 MiB
```

Remarquez la taille du pool dans la sortie précédente.

6. Augmentez la capacité de labpool en utilisant le disque /dev/vdc disponible dans le système.

- 6.1. Ajoutez le périphérique en mode bloc /dev/vdc à labpool à l'aide de la commande **stratis pool add-data**.

```
[root@serverb ~]# stratis pool add-data labpool /dev/vdc
```

- 6.2. Vérifiez la taille de labpool en utilisant la commande **stratis pool list**.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
labpool        10 GiB                56 MiB
```

La sortie précédente montre que la taille de labpool a augmenté après l'ajout d'un nouveau disque au pool.

- 6.3. Utilisez la commande **stratis blockdev list** pour lister les périphériques en mode bloc qui font désormais partie de labpool.

```
[root@serverb ~]# stratis blockdev list labpool  
Pool Name      Device Node    Physical Size   State   Tier  
labpool        /dev/vdb           5 GiB     In-use   Data  
labpool        /dev/vdc           5 GiB     In-use   Data
```

7. Créez un système de fichiers nommé **labfs** alloué de manière fine et dynamique dans le pool labpool. Montez ce système de fichiers sur **/labstratisvol** afin qu'il persiste lors

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

des redémarrages. Créez un fichier nommé **labfile1** contenant le texte **Hello World!** sur le système de fichiers **labfs**. N'oubliez pas d'utiliser l'option de montage **x-systemd.requires=stratisd.service** dans **/etc/fstab**.

- 7.1. Créez le système de fichiers nommé **labfs** alloué de manière fine et dynamique dans le pool **labpool** à l'aide de la commande **stratis filesystem create**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@serverb ~]# stratis filesystem create labpool labfs
```

- 7.2. Vérifiez la disponibilité de **labfs** en utilisant la commande **stratis filesystem list**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
                  UUID
labpool  labfs  546 MiB  Mar 29 2019 07:48  /stratis/labpool/labfs  9825...d6ca
```

Remarquez l'utilisation actuelle de **labfs**. Cette utilisation du système de fichiers augmente à la demande dans les étapes suivantes.

- 7.3. Déterminez l'UUID de **labfs** en utilisant la commande **lsblk**.

```
[root@serverb ~]# lsblk --output=UUID /stratis/labpool/labfs
UUID
9825e289-fb08-4852-8290-44d1b8f0d6ca
```

- 7.4. Modifiez **/etc/fstab** de sorte que le système de fichiers **labfs** alloué de manière fine et dynamique soit monté au moment du démarrage. Utilisez l'UUID que vous avez déterminé à l'étape précédente. Ce qui suit montre la ligne à ajouter à **/etc/fstab**. Vous pouvez utiliser la commande **vi /etc/fstab** pour éditer le fichier.

```
UUID=9825...d6ca /labstratisvol xfs defaults,x-systemd.requires=stratisd.service
0 0
```

- 7.5. Créez un répertoire nommé **/labstratisvol** en utilisant la commande **mkdir**.

```
[root@serverb ~]# mkdir /labstratisvol
```

- 7.6. Montez le système de fichiers **labfs** alloué de manière fine et dynamique en utilisant la commande **mount** pour confirmer que **/etc/fstab** contient les entrées appropriées.

```
[root@serverb ~]# mount /labstratisvol
```

Si la commande précédente produit des erreurs, revenez au fichier **/etc/fstab** et assurez-vous qu'il contient les entrées appropriées.

- 7.7. Créez un fichier texte nommé **/labstratisvol/labfile1** en utilisant la commande **echo**.

```
[root@serverb ~]# echo "Hello World!" > /labstratisvol/labfile1
```

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

8. Vérifiez que le système de fichiers `labfs` alloué de manière fine et dynamique croît réellement de manière dynamique à mesure que les données sur le système de fichiers augmentent.

- 8.1. Consultez l'utilisation actuelle de `labfs` en utilisant la commande **`stratis filesystem list`**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used       Created        Device
                    UUID
labpool  labfs  546 MiB  Mar 29 2019 07:48  /stratis/labpool/labfs  9825...d6ca
```

- 8.2. Créez un fichier de 2 Gio dans `labfs` en utilisant la commande **`dd`**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@serverb ~]# dd if=/dev/urandom of=/labstratisvol/labfile2 bs=1M count=2048
```

- 8.3. Vérifiez que l'utilisation de `labfs` a augmenté à l'aide de la commande **`stratis filesystem list`**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used       Created        Device
                    UUID
labpool  labfs  2.53 GiB  Mar 29 2019 07:48  /stratis/labpool/labfs  9825...d6ca
```

9. Créez un instantané nommé `labfs-snap` du système de fichiers `labfs`. L'instantané vous permet d'accéder à n'importe quel fichier supprimé de `labfs`.

- 9.1. Créez un instantané de `labfs` en utilisant la commande **`stratis filesystem snapshot`**. La commande peut prendre jusqu'à une minute pour s'exécuter.

```
[root@serverb ~]# stratis filesystem snapshot labpool \
labfs labfs-snap
```

- 9.2. Vérifiez la disponibilité de l'instantané en utilisant la commande **`stratis filesystem list`**.

```
[root@serverb ~]# stratis filesystem list
...output omitted...
labpool  labfs-snap  2.53 GiB  Mar 29 2019 10:28  /stratis/labpool/labfs-snap
291d...8a16
```

- 9.3. Supprimez le fichier `/labstratisvol/labfile1`.

```
[root@serverb ~]# rm /labstratisvol/labfile1
rm: remove regular file '/labstratisvol/labfile1'? y
```

- 9.4. Créez le répertoire `/labstratisvol-snap` en utilisant la commande **`mkdir`**.

```
[root@serverb ~]# mkdir /labstratisvol-snap
```

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

- 9.5. Montez l'instantané labfs-snap sur **/labstratisvol-snap** en utilisant la commande **mount**.

```
[root@serverb ~]# mount /stratis/labpool/labfs-snap \
/stratisvol-snap
```

- 9.6. Vérifiez que vous pouvez toujours accéder au fichier que vous avez supprimé de labfs en utilisant l'instantané labfs-snap.

```
[root@serverb ~]# cat /stratisvol-snap/labfile1
Hello World!
```

10. Créez le volume VDO labvdo, avec /dev/vdd comme périphérique. Définissez sa taille logique sur **50 Go**.

- 10.1. Créez le volume labvdo à l'aide de la commande **vdo create**.

```
[root@serverb ~]# vdo create --name=labvdo --device=/dev/vdd --vdoLogicalSize=50G
...output omitted...
```

- 10.2. Vérifiez la disponibilité du volume labvdo en utilisant la commande **vdo list**.

```
[root@serverb ~]# vdo list
labvdo
```

11. Montez le volume labvdo sur **/labvdovol** avec le système de fichiers XFS afin qu'il persiste lors des redémarrages. N'oubliez pas d'utiliser l'option de montage `x-systemd.requires=stratisd.service` dans **/etc/fstab**.

- 11.1. Formatez le volume labvdo avec le système de fichiers XFS en utilisant la commande **mkfs**.

```
[root@serverb ~]# mkfs.xfs -K /dev/mapper/labvdo
...output omitted...
```

- 11.2. Utilisez la commande **udevadm** pour enregistrer le nouveau noeud de périphérique.

```
[root@serverb ~]# udevadm settle
```

- 11.3. Créez le répertoire **/labvdovol** en utilisant la commande **mkdir**.

```
[root@serverb ~]# mkdir /labvdovol
```

- 11.4. Déterminez l'UUID de labvdo en utilisant la commande **lsblk**.

```
[root@serverb ~]# lsblk --output=UUID /dev/mapper/labvdo
UUID
ef8cce71-228a-478d-883d-5732176b39b1
```

- 11.5. Modifiez **/etc/fstab** pour que labvdo soit monté au moment du démarrage. Utilisez l'UUID du volume que vous avez déterminé à l'étape précédente. Ce qui suit montre la

**CHAPITRE 8 |** Mise en œuvre de fonctionnalités de stockage avancées

ligne à ajouter à **/etc/fstab**. Vous pouvez utiliser la commande **vi /etc/fstab** pour éditer le fichier.

```
UUID=ef8c...39b1 /labvdovol xfs defaults,x-systemd.requires=vdo.service 0 0
```

- 11.6. Montez le système de fichiers **labvdo** alloué de manière fine et dynamique en utilisant la commande **mount** pour confirmer que le fichier **/etc/fstab** contient les entrées appropriées.

```
[root@serverb ~]# mount /labvdovol
```

Si la commande précédente produit des erreurs, revenez au fichier **/etc/fstab** et assurez-vous qu'il contient les entrées appropriées.

12. Créez trois copies du fichier nommé **/root/install.img** sur le volume **labvdo**. Comparez les statistiques du volume pour vérifier la déduplication et la compression des données sur le volume.

- 12.1. Affichez les statistiques initiales et l'état du volume à l'aide de la commande **vdostats**.

```
[root@serverb ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.0G      2.0G  60%      99%
```

Notez que 3 Go du volume sont déjà utilisés, car lors de sa création, le volume VDO se réserve 3 à 4 Go pour lui-même. Notez également que la valeur **99%** du champ **Space saving%** indique que vous n'avez créé jusqu'à présent aucun contenu dans le volume, ce qui vous permet de disposer de tout l'espace de volume enregistré.

- 12.2. Copiez **/root/install.img** dans **/labvdovol/install.img.1** et vérifiez les statistiques du volume. La copie du fichier peut prendre jusqu'à une minute.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.1
[root@serverb ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.4G      1.6G  68%      5%
```

Notez que la valeur du champ **Used** est passée de **3.0G** à **3.4G**, car vous avez copié un fichier dans le volume et que cela occupe de l'espace. Notez également que la valeur du champ **Space saving%** est passée de **99%** à **5%**. En effet, le volume d'origine ne comportait aucun contenu, et cette valeur reflétait la faible utilisation de l'espace du volume et la disponibilité élevée de l'espace jusqu'à la création d'un fichier. La disponibilité de l'espace du volume est assez faible, car vous avez créé une copie unique du fichier dans le volume, et l'espace restant ne permet pas la déduplication.

- 12.3. Copiez **/root/install.img** dans **/labvdovol/install.img.2** et vérifiez les statistiques du volume. La copie du fichier peut prendre jusqu'à une minute.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.2
[root@serverb ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.4G      1.6G  68%      51%
```

Notez que l'espace de volume utilisé n'a pas changé. Au lieu de cela, le pourcentage de l'espace de volume enregistré a augmenté, prouvant que la déduplication des données s'est produite afin de réduire la consommation d'espace pour les copies redondantes du même fichier. La valeur de **Space saving%** dans la sortie précédente peut différer sur votre système.

12.4. Redémarrez la machine serverb.

```
[root@serverb ~]# systemctl reboot
```



### NOTE

**Remarque :** Si, au redémarrage, serverb ne démarre pas à partir d'une invite de connexion normale, mais affiche à la place le message « Give root password for maintenance (or press **Control-D** to continue): », vous avez probablement commis une erreur dans **/etc/fstab**. Après avoir fourni le mot de passe root de **chapeau rouge**, vous devrez remonter le système de fichiers racine en tant que read-write avec:

```
[root@serverb~]# monter -o remonter, rw /
```

Vérifiez que **/etc/fstab** est configuré correctement comme spécifié dans les solutions. Portez une attention particulière aux options de montage pour les lignes relatives à **/labstratisvol** et **/labvdovol**.

## Évaluation

Sur **workstation**, exécutez la commande **lab advstorage-review grade** pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab advstorage-review grade
```

## Terminer

Sur **workstation**, exécutez **lab advstorage-review finish** pour mettre fin à l'exercice. Ce script supprime les partitions et les fichiers créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab advstorage-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- La solution de gestion de stockage Stratis met en œuvre des systèmes de fichiers flexibles qui évoluent de manière dynamique avec les données.
- La solution de gestion de stockage Stratis prend en charge l'allocation fine et dynamique, la capture instantanée et la surveillance.
- Virtual Data Optimizer (VDO) vise à réduire les coûts de stockage des données.
- Virtual Data Optimizer applique l'élimination des blocs composés de zéros, la déduplication des données et la compression des données pour optimiser l'efficacité de l'espace disque.

## CHAPITRE 9

# ACCÈS AU STOCKAGE RATTACHÉ AU RÉSEAU

### PROJET

Accéder au stockage rattaché au réseau en utilisant le protocole NFS.

### OBJECTIFS

- Monter, utiliser et démonter une exportation NFS à partir de la ligne de commande et au démarrage.
- Configurer le service de montage automatique avec des schémas de correspondance directe et indirecte pour monter automatiquement un système de fichiers NFS à la demande et démonter ce dernier lorsqu'il n'est plus utilisé.
- Configurer un client NFS afin qu'il utilise NFSv4 à l'aide du nouvel outil **nfsconf**.

### SECTIONS

- Montage du stockage rattaché au réseau avec NFS (et exercice guidé)
- Montage automatique du stockage rattaché au réseau (et exercice guidé)

### ATELIER

Accès au stockage rattaché au réseau

# MONTAGE DU STOCKAGE RATTACHÉ AU RÉSEAU AVEC NFS

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Identifier les informations de partage NFS.
- Créer un répertoire à utiliser comme point de montage.
- Monter un partage NFS en utilisant la commande **mount** ou en configurant le fichier **/etc/fstab**.
- Démonter un partage NFS à l'aide de la commande **umount**.
- Configurer un client NFS afin qu'il utilise NFSv4 à l'aide du nouvel outil **nfsconf**.

## MONTAGE ET DÉMONTAGE DE PARTAGES NFS

NFS (*Network File System*) est un protocole Internet standard que Linux, UNIX et d'autres systèmes d'exploitation similaires utilisent comme système de fichiers en réseau natif. C'est une norme ouverte qui est encore en constante évolution et prend en charge les fonctions natives de Linux relatives aux permissions et aux systèmes de fichiers.

La version NFS par défaut en Red Hat Enterprise Linux 8 est 4.2. Les versions principales de NFSv4 et NFSv3 sont prises en charge. NFSv2 n'est plus pris en charge. NFSv4 communique uniquement avec le serveur par le biais du protocole TCP, tandis que les anciennes versions de NFS peuvent utiliser TCP ou UDP.

Les serveurs NFS exportent les partages (répertoires). Les clients NFS montent un partage exporté sur un point de montage local (répertoire) qui doit exister. Les partages NFS peuvent être montés de différentes façons :

- Manuellement, utilisez la commande **mount**.
- Automatiquement, au démarrage à l'aide d'entrées **/etc/fstab**.
- À la demande, en utilisant soit le service **autofs** ou les fonctions **systemd.automount**.

## Montage des partages NFS

Pour monter un partage NFS, suivez ces trois étapes :

1. **Identification** : l'administrateur du système client NFS peut identifier les partages NFS disponibles de différentes manières :

L'administrateur du serveur NFS peut fournir les détails d'exportation, et notamment les exigences de sécurité.

L'administrateur du système client peut également identifier les partages NFSv4 en montant le répertoire root du serveur NFS et en explorant les répertoires exportés. Pour ce faire, vous devez être connecté en tant qu'utilisateur **root**. L'accès aux partages qui utilisent la sécurité

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

Kerberos est refusé, mais le nom du partage (un répertoire) est visible. Vous pouvez parcourir les autres répertoires partagés.

```
[user@host ~]$ sudo mkdir mountpoint  
[user@host ~]$ sudo mount server:/ mountpoint  
[user@host ~]$ sudo ls mountpoint
```

2. **Point de montage** : utilisez **mkdir** pour créer un point de montage à un emplacement adapté.

```
[user@host ~]$ mkdir -p mountpoint
```

3. **Montage** : comme avec les systèmes de fichiers sur les partitions, les partages NFS doivent être montés pour être disponibles. Pour monter un partage NFS, sélectionnez l'une des options suivantes. Dans chaque cas, vous devez exécuter ces commandes en tant que superutilisateur, en vous connectant en tant que **root** ou en utilisant la commande **sudo**.
  - Montage temporaire : montez le partage NFS en utilisant la commande **mount** :

```
[user@host ~]$ sudo mount -t nfs -o rw,sync server:/share mountpoint
```

L'option **-t nfs** indique le type de système de fichiers des partages NFS (pas indispensable, mentionné à titre informatif). L'option **-o sync** oblige **mount** à synchroniser immédiatement les opérations d'écriture avec le serveur NFS (le mode est asynchrone par défaut).

Cette commande monte le partage immédiatement mais pas de manière persistante ; lors du prochain démarrage du système, ce partage NFS n'est pas disponible. Ceci est utile pour l'accès ponctuel aux données. Il est également utile de tester le montage d'un partage NFS avant de rendre le partage disponible de manière persistante.

- Montage de manière persistante : pour garantir que le partage NFS est monté au démarrage, modifiez le fichier **/etc/fstab** pour ajouter l'entrée de montage.

```
[user@host ~]$ sudo vim /etc/fstab  
...  
server:/share /mountpoint nfs rw,soft 0 0
```

Montez ensuite le partage NFS :

```
[user@host ~]$ sudo mount /mountpoint
```

Dans la mesure où le service client NFS trouve le serveur NFS et les options de montage dans le fichier **/etc/fstab**, vous n'avez pas besoin de les spécifier sur la ligne de commande.

## Démontage des partages NFS

En tant que **root** (ou en utilisant **sudo**), démontez un partage NFS à l'aide de la commande **umount**.

```
[user@host ~]$ sudo umount mountpoint
```

**NOTE**

Le démontage d'un partage n'entraîne pas la suppression de son entrée **/etc/fstab**. Sauf si vous supprimez ou commentez l'entrée, le partage NFS est remonté lors du prochain démarrage du système ou au redémarrage du service client NFS.

## L'OUTIL **nfsconf**

Red Hat Enterprise Linux 8 comprend désormais l'outil **nfsconf** pour gérer les fichiers de configuration client et serveur NFS sous NFSv4 et NFSv3. Configurez l'outil **nfsconf** à l'aide de **/etc/nfs.conf** (le fichier **/etc/sysconfig/nfs** des versions précédentes du système d'exploitation est à présent obsolète). Utilisez l'outil **nfsconf** pour obtenir, définir ou annuler la définition des paramètres de configuration NFS.

Le fichier de configuration **/etc/nfs.conf** est composé de plusieurs sections commençant par un mot-clé entre crochets. (**[mot-clé]**) avec des affectations de valeur dans la section. Pour un serveur NFS, configurez la section **[nfsd]**. Une affectation de valeur ou une clé est composée d'un nom pour la valeur, d'un signe égal et d'un paramètre pour la valeur, tel que **vers4.2=y**. Les lignes commençant par « # » ou « ; » sont ignorées, de même que les lignes vides.

```
[user@host ~]$ sudo cat /etc/nfs.conf
...output omitted...
[nfsd]
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# tcp=y
# vers2=n
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
# rdma=n
#
...output omitted...
```

Par défaut, les paires de clés de la section **[nfsd]** sont commentées. Toutefois, les commentaires indiquent les options par défaut qui entrent en vigueur si elles ne sont pas modifiées. Cela vous fournit un bon point de départ pour la configuration NFS.

Utilisez **nfsconf --set section key value** pour définir une valeur pour la clé de la section spécifiée.

```
[user@host ~]$ sudo nfsconf --set nfsd vers4.2 y
```

Cette commande met à jour le fichier de configuration **/etc/nfs.conf**:

```
[user@host ~]$ sudo cat /etc/nfs.conf
...output omitted...
[nfsd]
vers4.2 = y
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# tcp=y
# vers2=n
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
# rdma=n
#
...output omitted...
```

Utilisez **nfsconf --get section key** pour récupérer la valeur pour la clé de la section spécifiée :

```
[user@host ~]$ sudo nfsconf --get nfsd vers4.2
y
```

Utilisez **nfsconf --unset section key** pour annuler la définition de la valeur pour la clé de la section spécifiée :

```
[user@host ~]$ sudo nfsconf --unset nfsd vers4.2
```

## Configurer un client uniquement NFSv4

Vous pouvez configurer un client uniquement NFSv4 en définissant les valeurs suivantes dans le fichier de configuration **/etc/nfs.conf**.

Commencez par désactiver UDP et les autres clés liées à NFSv2 et NFSv3 :

```
[user@host ~]$ sudo nfsconf --set nfsd udp n
[user@host ~]$ sudo nfsconf --set nfsd vers2 n
[user@host ~]$ sudo nfsconf --set nfsd vers3 n
```

Activez TCP et les clés liées à NFSv4.

```
[user@host ~]$ sudo nfsconf --set nfsd tcp y
[user@host ~]$ sudo nfsconf --set nfsd vers4 y
[user@host ~]$ sudo nfsconf --set nfsd vers4.0 y
[user@host ~]$ sudo nfsconf --set nfsd vers4.1 y
[user@host ~]$ sudo nfsconf --set nfsd vers4.2 y
```

Comme auparavant, les modifications apparaissent dans le fichier de configuration **/etc/nfs.conf**:

```
[[user@host ~]$ cat /etc/nfs.conf
[nfsd]
udp = n
vers2 = n
vers3 = n
tcp = y
vers4 = y
vers4.0 = y
vers4.1 = y
vers4.2 = y
...output omitted...
```



## RÉFÉRENCES

Pages de manuel **mount(8)**, **umount(8)**, **fstab(5)**, **mount.nfs(8)**, **nfs.conf(8)** and **nfsconf(8)**

## ► EXERCICE GUIDÉ

# GESTION DU STOCKAGE RATTACHÉ AU RÉSEAU AVEC NFS

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet exercice, vous allez modifier le fichier **/etc/fstab** pour monter de manière persistante une exportation NFS au démarrage.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Tester un serveur NFS en utilisant la commande **mount**.
- Configurer les partages NFS dans le fichier de configuration **/etc/fstab** pour enregistrer les modifications même après le redémarrage du système.
- Configurer les clients NFS afin qu'ils utilisent NFSv4 à l'aide du nouvel outil **nfsconf**.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab netstorage-nfs start**. Cette commande exécute un script de démarrage qui détermine si les machines **servera** et **serverb** sont accessibles sur le réseau. Le script vous alerte si elles ne sont pas disponibles. Le script de démarrage configure **serverb** en tant que serveur NFSv4, configure les permissions et exporte les répertoires. Il crée des utilisateurs et des groupes nécessaires sur **servera** et **serverb**.

```
[student@workstation ~]$ lab netstorage-nfs start
```

Une entreprise de transport utilise un serveur central, **serverb**, pour héberger un certain nombre de documents et de répertoires partagés. Les utilisateurs sur **servera**, qui sont tous membres du groupe **admin**, ont besoin d'accéder au partage NFS monté de manière persistante.

Remarques importantes :

- **serverb** partage le répertoire **/shares/public** qui contient des fichiers texte.
  - Les membres du groupe **admin** (**admin1**, **sysmanager1**) ont un accès en lecture et en écriture au répertoire partagé **/shares/public**.
  - Le point de montage principal pour **servera** est **/public**.
  - Tous les utilisateurs ont le même mot de passe : **redhat**.
- 1. Connectez-vous à **servera** en tant qu'utilisateur **student** et basculez vers l'utilisateur **root**.
- 1.1. Connectez-vous à **servera** en tant qu'utilisateur **student**.

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur root. Le mot de passe de l'utilisateur student est student.

```
[student@servera ~]$ sudo -i  
[sudo] password for student:  
[root@servera ~]#
```

- ▶ 2. Utilisez l'outil **nfsconf** pour configurer **/etc/nfs.conf** afin d'activer le serveur NFS pour qu'il fonctionne dans la version 4.X. Assurez-vous également que le mode TCP est activé et que le mode UDP est désactivé.

- 2.1. Utilisez l'outil **nfsconf** pour désactiver les clés **udp**, **vers2** et **vers3**.

```
[root@servera ~]# nfsconf --set nfsd udp n  
[root@servera ~]# nfsconf --set nfsd vers2 n  
[root@servera ~]# nfsconf --set nfsd vers3 n
```

- 2.2. Utilisez l'outil **nfsconf** pour activer les clés **tcp**, **vers4**, **vers4.0**, **vers4.1**, **vers4.2**.

```
[root@servera ~]# nfsconf --set nfsd tcp y  
[root@servera ~]# nfsconf --set nfsd vers4 y  
[root@servera ~]# nfsconf --set nfsd vers4.0 y  
[root@servera ~]# nfsconf --set nfsd vers4.1 y  
[root@servera ~]# nfsconf --set nfsd vers4.2 y
```

- ▶ 3. Testez le serveur NFS sur serverb en utilisant servera en tant que client NFS.

- 3.1. Créez le point de montage **/public** sur servera.

```
[root@servera ~]# mkdir /public
```

- 3.2. Sur servera, utilisez la commande **mount** pour vérifier que le partage NFS **/share/public** exporté par serverb se monte correctement sur le point de montage **/public**.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares/public /public
```

- 3.3. Listez le contenu du partage NFS monté.

```
[root@servera ~]# ls -l /public  
total 16  
-rw-r--r--. 1 root admin 42 Apr  8 22:36 Delivered.txt  
-rw-r--r--. 1 root admin 46 Apr  8 22:36 NOTES.txt  
-rw-r--r--. 1 root admin 20 Apr  8 22:36 README.txt  
-rw-r--r--. 1 root admin 27 Apr  8 22:36 Trackings.txt
```

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

3.4. Explorez les options **mount** pour le partage NFS monté.

```
[root@servera ~]# mount | grep public
serverb.lab.example.com:/shares/public on /public type nfs4
(rw,relatime,vers=4.2,rsiz...
```

3.5. Démontez le partage NFS.

```
[root@servera ~]# umount /public
```

► 4. Configurez **servera** pour garantir que le partage utilisé ci-dessus est monté de manière persistante.

4.1. Ouvrez le fichier **/etc/fstab** pour l'éditer.

```
[root@servera ~]# vim /etc/fstab
```

Ajoutez la ligne suivante à la fin du fichier :

```
serverb.lab.example.com:/shares/public /public nfs rw,sync 0 0
```

4.2. Utilisez la commande **mount** pour monter le répertoire partagé.

```
[root@servera ~]# mount /public
```

4.3. Listez le contenu du répertoire partagé.

```
[root@servera ~]# ls -l /public
total 16
-rw-r--r-- 1 root    admin 42 Apr  8 22:36 Delivered.txt
-rw-r--r-- 1 root    admin 46 Apr  8 22:36 NOTES.txt
-rw-r--r-- 1 root    admin 20 Apr  8 22:36 README.txt
-rw-r--r-- 1 root    admin 27 Apr  8 22:36 Trackings.txt
```

4.4. Redémarrez la machine **servera**.

```
[root@servera ~]# systemctl reboot
```

► 5. Une fois le redémarrage de **servera** terminé, connectez-vous à **servera** en tant qu'utilisateur **admin1** et testez le partage NFS monté de manière persistante.

5.1. Connectez-vous à **servera** en tant qu'utilisateur **admin1**.

```
[student@workstation ~]$ ssh admin1@servera
[admin1@servera ~]$
```

5.2. Testez le partage NFS monté sur **/public**

```
[admin1@servera ~]$ ls -l /public
total 16
-rw-r--r--. 1 root    admin 42 Apr  8 22:36 Delivered.txt
-rw-r--r--. 1 root    admin 46 Apr  8 22:36 NOTES.txt
-rw-r--r--. 1 root    admin 20 Apr  8 22:36 README.txt
-rw-r--r--. 1 root    admin 27 Apr  8 22:36 Trackings.txt
[admin1@servera ~]$ cat /public/NOTES.txt
###In this file you can log all your notes###
[admin1@servera ~]$ echo "This is a test" > /public/Test.txt
[admin1@servera ~]$ cat /public/Test.txt
This is a test
```

5.3. Déconnectez-vous de servera.

```
[admin1@servera ~]$ exit
logout
Connection to servera closed.
```

## Fin

Sur workstation, exéutez le script **lab netstorage-nfs finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netstorage-nfs finish
```

L'exercice guidé est maintenant terminé.

# MONTAGE AUTOMATIQUE DU STOCKAGE RATTACHÉ AU RÉSEAU

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Décrire les avantages liés à l'utilisation du service de montage automatique.
- Monter automatiquement des partages NFS à l'aide de schémas de correspondance directe et indirecte incluant des caractères génériques.

## MONTAGE DE PARTAGES NFS AVEC LE SERVICE DE MONTAGE AUTOMATIQUE

Le service de *montage automatique* (`autofs`) monte automatiquement des partages NFS « à la demande », et démonte automatiquement les partages NFS qui ne sont plus utilisés.

### Avantages du service de montage automatique

- L'utilisateur n'a pas besoin de privilèges root pour exécuter les commandes `mount` et `umount`.
- Les partages NFS configurés dans le service de montage automatique sont accessibles à tous les utilisateurs de la machine, si leurs permissions d'accès les y autorisent.
- Les partages NFS ne sont pas connectés de façon permanente comme les entrées du fichier / **etc/fstab**, ce qui libère des ressources réseau et système.
- Le service de montage automatique est configuré côté client. Aucune configuration n'est requise sur le serveur.
- Le service de montage automatique utilise les mêmes options que la commande `mount`, dont les options de sécurité.
- Le service de montage automatique prend en charge la mise en correspondance des points de montage directs et indirects, offrant ainsi une grande souplesse au niveau des emplacements de point de montage.
- `autofs` crée et supprime des points de montage indirects, éliminant ainsi la gestion manuelle.
- NFS est le système de fichiers réseau du service de montage automatique par défaut, mais d'autres systèmes de fichiers réseau peuvent être montés automatiquement.
- `autofs` est un service géré de la même façon que les autres services du système.

### Création d'un montage automatique

La configuration d'un montage automatique se déroule en plusieurs étapes :

1. Installez le paquetage `autofs`.

```
[user@host ~]$ sudo yum install autofs
```

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

Ce paquetage contient tous les éléments nécessaires à l'utilisation du service de montage automatique pour les partages NFS.

2. Ajoutez un fichier de schéma de correspondance maître à **/etc/auto.master.d**. Ce fichier identifie le répertoire de base utilisé pour les points de montage, ainsi que le fichier de mise en correspondance utilisé pour créer des montages automatiques.

```
[user@host ~]$ sudo vim /etc/auto.master.d/demo.autofs
```

Le nom du fichier de schéma de correspondance maître est arbitraire (bien qu'il soit généralement en rapport avec son contenu), mais il doit avoir l'extension **.autofs** pour que le sous-système le reconnaisse. Vous pouvez placer plusieurs entrées dans un seul fichier de schéma de correspondance maître. Vous pouvez également créer plusieurs fichiers de schéma de correspondance maître, chacun avec ses propres entrées regroupées de manière logique.

Ajoutez l'entrée du schéma de correspondance maître. Dans notre exemple, elle s'applique aux montages à mise en correspondance indirecte :

```
/shares  /etc/auto.demo
```

Cette entrée utilise le répertoire **/shares** comme base pour les futurs montages automatiques indirects. Le fichier **/etc/auto.demo** contient les détails du montage. Utilisez un nom de fichier absolu. Le fichier **auto.demo** doit être créé avant le démarrage du service **autofs**.

3. Créez les fichiers de mise en correspondance. Chaque fichier de mise en correspondance identifie le point de montage, les options de montage, ainsi que l'emplacement source à monter pour un ensemble de montages automatiques.

```
[user@host ~]$ sudo vim /etc/auto.demo
```

Par convention, le nom d'un fichier de mise en correspondance comporte **/etc/auto.name**, où le *nom* reflète le contenu du schéma de correspondance.

```
work  -rw, sync  server:/shares/work
```

Le format d'une entrée est *point de montage, options de montage et emplacement source*. Cet exemple présente une entrée de mise en correspondance indirecte de base. Les schémas de correspondance directe et indirecte qui utilisent des caractères génériques seront abordés plus loin dans cette section.

- Connue sous le nom de « key » dans les pages de manuel, le *point de montage* est créé et supprimé automatiquement par le service **autofs**. Dans le cas présent, le nom complet du point de montage est **/shares/work** (voir le fichier de schéma de correspondance maître). Le répertoire **/shares** et les répertoires **/shares/work** seront créés et supprimés selon les besoins par le service **autofs**.

Dans cet exemple, le point de montage local reflète la structure de répertoire du serveur, mais cela n'est pas obligatoire. Le point de montage local peut porter n'importe quel nom. Le service **autofs** n'impose pas de structure conventionnelle de nom spécifique sur le client.

## CHAPITRE 9 | Accès au stockage rattaché au réseau

- Les options de montage commencent par un tiret (-) et sont séparées par des virgules, sans espace. Les options de montage disponibles pour le montage manuel d'un système de fichiers sont disponibles lors du montage automatique. Dans cet exemple, le service de montage automatique monte le partage avec accès en lecture/écriture (option **rw**), et le serveur est synchronisé immédiatement pendant les opérations d'écriture (option **sync**).

Les options utiles propres au service de montage automatique comprennent **-fstype=** et **-strict**. Utilisez **fstype** pour spécifier un type de système de fichiers, par exemple, **nfs4** ou **xfs**, et utilisez **strict** pour que les erreurs de montage de systèmes de fichiers soient traitées comme étant irrécupérables.

- L'emplacement source des partages NFS est indiqué sous la forme host : /pathname. Dans cet exemple, il s'agit de **&srvb ; :/shares/work**. Pour que ce montage automatique réussisse, le serveur NFS **serverb** doit exporter le répertoire avec accès en lecture/écriture, et l'utilisateur qui demande l'accès doit disposer de permissions de fichier Linux standard sur le répertoire. Si **serverb** exporte le répertoire avec l'accès en lecture seule, le client obtiendra l'accès en lecture seule même s'il a demandé l'accès en lecture/écriture.

### 4. Démarrez et activez le service de montage automatique.

Utilisez **systemctl** pour démarrer et activer le service **autofs**.

```
[user@host ~]$ sudo systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/
lib/systemd/system/autofs.service.
```

## Schémas de correspondance directe

Les schémas de correspondance directe servent à mettre en correspondance un partage NFS avec un point de montage existant comportant un chemin absolu.

Pour utiliser des points de montage directement mis en correspondance, le fichier de schéma de correspondance maître peut apparaître comme suit :

```
/ - /etc/auto.direct
```

Toutes les entrées de schéma de correspondance directe utilisent **/-** comme répertoire de base. Dans notre exemple, le fichier de mise en correspondance contenant les détails de montage est **/etc/auto.direct**.

Le contenu du fichier **/etc/auto.direct** peut se présenter comme suit :

```
/mnt/docs -rw, sync serverb:/shares/docs
```

Le point de montage (ou « key ») est toujours un chemin absolu. Le reste du fichier de mise en correspondance utilise la même structure.

Dans cet exemple, le répertoire **/mnt** existe, et n'est pas géré par **autofs**. Le répertoire entier **/mnt/docs** est créé et supprimé automatiquement par le service **autofs**.

## Schémas de correspondance indirecte avec caractères génériques

Lorsqu'un serveur NFS exporte plusieurs sous-répertoires d'un même répertoire, le service de montage automatique peut être configuré pour accéder à l'ensemble de ces sous-répertoires avec une seule entrée de mise en correspondance.

Pour poursuivre l'exemple précédent, si `&svrb;:/shares` exporte au moins deux sous-répertoires accessibles à l'aide des mêmes options de montage, le contenu du fichier `/etc/auto.demo` peut alors s'afficher comme suit :

```
* -rw, sync serverb:/shares/&
```

Le point de montage (ou « key ») est un astérisque (\*) et le sous-répertoire à l'emplacement source est une esperluette (&). Tout le reste de l'entrée est identique.

Lorsqu'un utilisateur tente d'accéder à `/shares/work`, le point de montage \* (qui correspond à `work` dans cet exemple) remplace l'esperluette de l'emplacement source, et le partage `&svrb;:/shares/work` est monté. Comme dans l'exemple de mise en correspondance indirecte, le répertoire `work` est créé et supprimé automatiquement par le service `autofs`.



### RÉFÉRENCES

Pages du manuel **autofs(5)**, **automount(8)**, **auto.master(5)** et **mount.nfs(8)**

## ► EXERCICE GUIDÉ

# MONTAGE AUTOMATIQUE DU STOCKAGE RATTACHÉ AU RÉSEAU

### LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet exercice, vous allez créer des points de montage gérés par le service de montage automatique pour les mises en correspondance directe et indirecte qui montent des systèmes de fichiers NFS.

### RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Installer les paquetages requis pour le service de montage automatique.
- Configurer les schémas de correspondance directe et indirecte du service de montage automatique, en obtenant des ressources d'un serveur NFSv4 préconfiguré.
- Comprendre les différences entre le schéma de correspondance directe et le schéma de correspondance indirecte.

### AVANT DE COMMENCER

Connectez-vous à `workstation` en tant que `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab netstorage-autofs start`. Cette commande exécute un script de démarrage qui détermine si `servera` et `serverb` sont accessibles sur le réseau. Le script vous alerte s'ils ne sont pas disponibles. Le script de démarrage configure `serverb` en tant que serveur NFSv4, configure les permissions et exporte les répertoires. Il crée également des utilisateurs et des groupes nécessaires sur `servera` et `serverb`.

```
[student@workstation ~]$ lab netstorage-autofs start
```

Un fournisseur de services Internet utilise un serveur central, `serverb`, pour héberger des répertoires partagés qui contiennent des documents importants devant être disponibles à la demande. Lorsque les utilisateurs se connectent à `servera`, ils ont besoin d'accéder aux répertoires partagés montés automatiquement.

Remarques importantes :

- `serverb` exporte le répertoire `/shares/indirect` en tant que partage NFS, qui contient tous les sous-répertoires `west`, `central` et `east`.
- `serverb` exporte également le répertoire `/shares/direct/external` en tant que partage NFS.
- Le groupe `operators` est constitué des utilisateurs `operator1` et `operator2`. Ils ont un accès en lecture et en écriture aux répertoires partagés `/shares/indirect/west`, `/shares/indirect/central` et `/shares/indirect/east`.

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

- Le groupe **contractors** se compose des utilisateurs **contractor1** et **contractor2**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/direct/external**.
- Les points de montage attendus pour **servera** sont **/external** et **/internal**.
- Le répertoire partagé **/shares/direct/external** doit être monté automatiquement sur **servera** à l'aide d'un schéma de correspondance *directe* sur **/external**.
- Le répertoire partagé **/shares/indirect/west** doit être monté automatiquement sur **servera** à l'aide d'un schéma de correspondance *indirecte* sur **/internal/west**.
- Le répertoire partagé **/shares/indirect/central** doit être monté automatiquement sur **servera** à l'aide d'un schéma de correspondance *indirecte* sur **/internal/central**.
- Le répertoire partagé **/shares/indirect/east** doit être monté automatiquement sur **servera** à l'aide d'un schéma de correspondance *indirecte* sur **/internal/east**.
- Tous les utilisateurs ont le même mot de passe : **redhat**.

► 1. Connectez-vous à **servera** et installez les paquetages requis.

1.1. Connectez-vous à **servera** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

1.3. Installez le paquetage **autofs**.

```
[root@servera ~]# yum install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
```

► 2. Configurez un schéma de correspondance directe du service de montage automatique sur **servera** en utilisant des partages de **serverb**. Créez le schéma de correspondance directe en utilisant les fichiers nommés **/etc/auto.master.d/direct.autofs** pour le schéma de correspondance maître et **/etc/auto.direct** pour le fichier de mise en correspondance. Utilisez le répertoire **/external** en tant que point de montage principal sur **servera**.

2.1. Testez le serveur et le partage NFS avant de configurer le service de montage automatique.

```
[root@servera ~]# mount -t nfs \
serverb.lab.example.com:/shares/direct/external /mnt
[root@servera ~]# ls -l /mnt
total 4
-rw-r--r--. 1 root contractors 22 Apr 7 23:15 README.txt
[root@servera ~]# umount /mnt
```

- 2.2. Créez un fichier de schéma de correspondance maître nommé **/etc/auto.master.d/direct.autofs**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.master.d/direct.autofs
/- /etc/auto.direct
```

- 2.3. Créez un fichier de schéma de correspondance directe nommé **/etc/auto.direct**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.direct
/external -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/direct/external
```

- 3. Configurez un schéma de correspondance indirecte du service de montage automatique sur **servera** en utilisant des partages de **serverb**. Créez le schéma de correspondance indirecte en utilisant les fichiers nommés **/etc/auto.master.d/indirect.autofs** pour le schéma de correspondance maître et **/etc/auto.indirect** pour le fichier de mise en correspondance. Utilisez le répertoire **/internal** en tant que point de montage principal sur **servera**.

- 3.1. Testez le serveur et le partage NFS avant de configurer le service de montage automatique.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares/indirect /mnt
[root@servera ~]# ls -l /mnt
total 0
drwxrws---. 2 root operators 24 Apr 7 23:34 central
drwxrws---. 2 root operators 24 Apr 7 23:34 east
drwxrws---. 2 root operators 24 Apr 7 23:34 west
[root@servera ~]# umount /mnt
```

- 3.2. Créez un fichier de schéma de correspondance maître nommé **/etc/auto.master.d/indirect.autofs**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.master.d/indirect.autofs
/internal /etc/auto.indirect
```

- 3.3. Créez un fichier de schéma de correspondance indirecte nommé **/etc/auto.indirect**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.indirect
* -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/indirect/&
```

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

- 4. Activez le service `autofs` sur `servera` et activez-le automatiquement au démarrage. Redémarrez `servera` pour déterminer si le service `autofs` démarre automatiquement.

4.1. Lancez et activez le service `autofs` sur `servera`.

```
[root@servera ~]# systemctl enable --now autofs  
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/  
lib/systemd/system/autofs.service.
```

4.2. Redémarrez la machine `servera`.

```
[root@servera ~]# systemctl reboot
```

- 5. Testez le schéma de correspondance directe du service de montage automatique en tant qu'utilisateur `contractor1`. Lorsque vous avez terminé, quittez la session utilisateur `contractor1` sur `servera`.

5.1. Après le redémarrage de la machine `servera`, connectez-vous à `servera` en tant qu'utilisateur `student`.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

5.2. Basculez vers l'utilisateur `contractor1`.

```
[student@servera ~]$ su - contractor1  
Password: redhat
```

5.3. Listez le point de montage `/external`.

```
[contractor1@servera ~]$ ls -l /external  
total 4  
-rw-r--r--. 1 root contractors 22 Apr 7 23:34 README.txt
```

5.4. Passez en revue le contenu et testez l'accès au point de montage `/external`.

```
[contractor1@servera ~]$ cat /external/README.txt  
###External Folder###  
[contractor1@servera ~]$ echo testing-direct > /external/testing.txt  
[contractor1@servera ~]$ cat /external/testing.txt  
testing-direct
```

5.5. Quittez la session utilisateur `contractor1`.

```
[contractor1@servera ~]$ exit  
logout
```

- 6. Testez le schéma de correspondance indirecte du service de montage automatique en tant qu'utilisateur `operator1`. Lorsque vous avez terminé, déconnectez-vous de `servera`.

## 6.1. Basculez vers l'utilisateur operator1.

```
[student@servera ~]$ su - operator1
Password: redhat
```

## 6.2. Listez le point de montage /internal.

```
[operator1@servera ~]$ ls -l /internal
total 0
```

**NOTE**

Vous remarquerez que dans un schéma de correspondance indirecte du service de montage automatique, même si vous êtes au point de montage mis en correspondance, vous devez appeler chacun des sous-répertoires partagés ou des fichiers à la demande pour pouvoir y accéder. Dans un schéma de correspondance directe du service de montage automatique, après avoir ouvert le point de montage mis en correspondance, vous avez accès aux répertoires et au contenu configurés dans le répertoire partagé.

## 6.3. Testez l'accès au répertoire partagé du service de montage automatique /internal/west.

```
[operator1@servera ~]$ ls -l /internal/west/
total 4
-rw-r--r-- 1 root operators 18 Apr  7 23:34 README.txt
[operator1@servera ~]$ cat /internal/west/README.txt
###West Folder###
[operator1@servera ~]$ echo testing-1 > /internal/west/testing-1.txt
[operator1@servera ~]$ cat /internal/west/testing-1.txt
testing-1
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws--- 2 root operators 24 Apr  7 23:34 west
```

## 6.4. Testez l'accès au répertoire partagé du service de montage automatique /internal/central.

```
[operator1@servera ~]$ ls -l /internal/central
total 4
-rw-r--r-- 1 root operators 21 Apr  7 23:34 README.txt
[operator1@servera ~]$ cat /internal/central/README.txt
###Central Folder###
[operator1@servera ~]$ echo testing-2 > /internal/central/testing-2.txt
[operator1@servera ~]$ cat /internal/central/testing-2.txt
testing-2
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws--- 2 root operators 24 Apr  7 23:34 central
drwxrws--- 2 root operators 24 Apr  7 23:34 west
```

6.5. Testez l'accès au répertoire partagé du service de montage automatique **/internal/east**.

```
[operator1@servera ~]$ ls -l /internal/east
total 4
-rw-r--r--. 1 root operators 18 Apr  7 23:34 README.txt
[operator1@servera ~]$ cat /internal/east/README.txt
###East Folder###
[operator1@servera ~]$ echo testing-3 > /internal/east/testing-3.txt
[operator1@servera ~]$ cat /internal/east/testing-3.txt
testing-3
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws---. 2 root operators 24 Apr  7 23:34 central
drwxrws---. 2 root operators 24 Apr  7 23:34 east
drwxrws---. 2 root operators 24 Apr  7 23:34 west
```

6.6. Testez l'accès au répertoire partagé du service de montage automatique **/external**.

```
[operator1@servera ~]$ ls -l /external
ls: cannot open directory '/external': Permission denied
```

6.7. Déconnectez-vous de servera.

```
[operator1@servera ~]$ exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
```

## Fin

Sur workstation, exécutez le script **lab netstorage-autofs finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netstorage-autofs finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# ACCÈS AU STOCKAGE RATTACHÉ AU RÉSEAU

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez configurer le service de montage automatique avec un schéma de correspondance indirecte, à l'aide des partages d'un serveur NFSv4.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Installer les paquetages requis pour configurer le service de montage automatique.
- Configurer un schéma de correspondance indirecte du service de montage automatique, en obtenant des ressources d'un serveur NFSv4 préconfiguré.
- Configurer un client NFS afin qu'il utilise NFSv4 à l'aide de l'outil **nfsconf**.

## AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab netstorage-review start**. Celle-ci exécute un script de démarrage qui détermine si les systèmes **servera** et **serverb** sont accessibles sur le réseau. Le script de démarrage configure **serverb** en tant que serveur NFSv4, configure les permissions et exporte les répertoires. Il crée également des utilisateurs et des groupes nécessaires sur les systèmes **servera** et **serverb**.

```
[student@workstation ~]$ lab netstorage-review start
```

Une entreprise de support informatique utilise un serveur central, **serverb**, pour héberger des répertoires partagés sur **/remote/shares** pour leurs groupes et utilisateurs. Les utilisateurs doivent pouvoir se connecter et disposer de leurs répertoires partagés montés à la demande et prêts à être utilisés, sous le répertoire **/shares** sur **servera**.

Remarques importantes :

- **serverb** partage le répertoire **/shares** qui contient tous les sous-répertoires **management**, **production** et **operation**.
- Le groupe **managers** se compose des utilisateurs **manager1** et **manager2**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/management**.
- Le groupe **production** se compose des utilisateurs **dbuser1** et **sysadmin1**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/production**.
- Le groupe **operators** se compose des utilisateurs **contractor1** et **consultant1**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/operation**.
- Le point de montage principal de **servera** est le répertoire **/remote**.

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

- Le répertoire partagé **/shares/management** doit être monté automatiquement sur **/remote/management** et sur **servera**.
  - Le répertoire partagé **/shares/production** doit être monté automatiquement sur **/remote/production** et sur **servera**.
  - Le répertoire partagé **/shares/operation** doit être monté automatiquement sur **/remote/operation** et sur **servera**.
  - Tous les utilisateurs ont le même mot de passe : **redhat**.
1. Connectez-vous à **servera** et installez les paquetages requis.
  2. Utilisez la commande **nfsconf** pour configurer **/etc/nfs.conf**. Activez le client NFS pour qu'il fonctionne uniquement dans la version 4.X. Assurez-vous également que le mode TCP est activé et que le mode UDP est désactivé.
  3. Configurez un schéma de correspondance indirecte du service de montage automatique sur **servera** en utilisant des partages de **Serverb**. Créez un schéma de correspondance indirecte en utilisant les fichiers nommés **/etc/auto.master.d/shares.autofs** pour le schéma de correspondance maître et **/etc/auto.shares** pour le fichier de mise en correspondance. Utilisez le répertoire **/remote** en tant que point de montage principal sur **servera**. Redémarrez **servera** pour déterminer si le service **autofs** démarre automatiquement.
  4. Testez la configuration **autofs** avec les différents utilisateurs. Lorsque vous avez terminé, déconnectez-vous de **servera**.

## Évaluation

À partir de **workstation**, exécutez la commande **lab netstorage-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab netstorage-review grade
```

## Fin

Sur **workstation**, exécutez la commande **lab netstorage-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netstorage-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# ACCÈS AU STOCKAGE RATTACHÉ AU RÉSEAU

### LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez configurer le service de montage automatique avec un schéma de correspondance indirecte, à l'aide des partages d'un serveur NFSv4.

### RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Installer les paquetages requis pour configurer le service de montage automatique.
- Configurer un schéma de correspondance indirecte du service de montage automatique, en obtenant des ressources d'un serveur NFSv4 préconfiguré.
- Configurer un client NFS afin qu'il utilise NFSv4 à l'aide de l'outil **nfsconf**.

### AVANT DE COMMENCER

Connectez-vous à **workstation** en tant que **student** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab netstorage-review start**. Celle-ci exécute un script de démarrage qui détermine si les systèmes **servera** et **serverb** sont accessibles sur le réseau. Le script de démarrage configure **serverb** en tant que serveur NFSv4, configure les permissions et exporte les répertoires. Il crée également des utilisateurs et des groupes nécessaires sur les systèmes **servera** et **serverb**.

```
[student@workstation ~]$ lab netstorage-review start
```

Une entreprise de support informatique utilise un serveur central, **serverb**, pour héberger des répertoires partagés sur **/remote/shares** pour leurs groupes et utilisateurs. Les utilisateurs doivent pouvoir se connecter et disposer de leurs répertoires partagés montés à la demande et prêts à être utilisés, sous le répertoire **/shares** sur **servera**.

Remarques importantes :

- **serverb** partage le répertoire **/shares** qui contient tous les sous-répertoires **management**, **production** et **operation**.
- Le groupe **managers** se compose des utilisateurs **manager1** et **manager2**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/management**.
- Le groupe **production** se compose des utilisateurs **dbuser1** et **sysadmin1**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/production**.
- Le groupe **operators** se compose des utilisateurs **contractor1** et **consultant1**. Ces derniers ont un accès en lecture et en écriture au répertoire partagé **/shares/operation**.
- Le point de montage principal de **servera** est le répertoire **/remote**.

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

- Le répertoire partagé **/shares/management** doit être monté automatiquement sur **/remote/management** et sur **servera**.
- Le répertoire partagé **/shares/production** doit être monté automatiquement sur **/remote/production** et sur **servera**.
- Le répertoire partagé **/shares/operation** doit être monté automatiquement sur **/remote/operation** et sur **servera**.
- Tous les utilisateurs ont le même mot de passe : **redhat**.

**1.** Connectez-vous à **servera** et installez les paquetages requis.1.1. Connectez-vous à **servera** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

1.2. Utilisez la commande **sudo -i** pour basculer vers l'utilisateur **root**. Le mot de passe de l'utilisateur **student** est **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

1.3. Installez le paquetage **autofs**.

```
[root@servera ~]# yum install autofs  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...
```

**2.** Utilisez la commande **nfsconf** pour configurer **/etc/nfs.conf**. Activez le client NFS pour qu'il fonctionne uniquement dans la version 4.X. Assurez-vous également que le mode TCP est activé et que le mode UDP est désactivé.2.1. Utilisez l'outil **nfsconf** pour désactiver les clés **udp**, **vers2** et **vers3**.

```
[root@servera ~]# nfsconf --set nfsd udp n  
[root@servera ~]# nfsconf --set nfsd vers2 n  
[root@servera ~]# nfsconf --set nfsd vers3 n
```

2.2. Utilisez l'outil **nfsconf** pour activer les clés **tcp**, **vers4**, **vers4.0**, **vers4.1**, **vers4.2**.

```
[root@servera ~]# nfsconf --set nfsd tcp y  
[root@servera ~]# nfsconf --set nfsd vers4 y  
[root@servera ~]# nfsconf --set nfsd vers4.0 y  
[root@servera ~]# nfsconf --set nfsd vers4.1 y  
[root@servera ~]# nfsconf --set nfsd vers4.2 y
```

**3.** Configurez un schéma de correspondance indirecte du service de montage automatique sur **servera** en utilisant des partages de **serverb**. Créez un schéma de correspondance

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

indirecte en utilisant les fichiers nommés **/etc/auto.master.d/shares.autofs** pour le schéma de correspondance maître et **/etc/auto.shares** pour le fichier de mise en correspondance. Utilisez le répertoire **/remote** en tant que point de montage principal sur **servera**. Redémarrez **servera** pour déterminer si le service **autofs** démarre automatiquement.

- 3.1. Testez le serveur avant de configurer le service de montage automatique.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares /mnt
[root@servera ~]# ls -l /mnt
total 0
drwxrwx--- 2 root managers 25 Apr  4 01:13 management
drwxrwx--- 2 root operators 25 Apr  4 01:13 operation
drwxrwx--- 2 root production 25 Apr  4 01:13 production
[root@servera ~]# umount /mnt
```

- 3.2. Créez un fichier de schéma de correspondance maître nommé **/etc/auto.master.d/shares.autofs**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.master.d/shares.autofs
/remote /etc/auto.shares
```

- 3.3. Créez un fichier de schéma de correspondance indirecte nommé **/etc/auto.shares**, insérez le contenu suivant, puis enregistrez les modifications.

```
[root@servera ~]# vim /etc/auto.shares
* -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/&
```

- 3.4. Lancez et activez le service **autofs** sur **servera**.

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/
lib/systemd/system/autofs.service.
```

- 3.5. Redémarrez la machine **servera**.

```
[root@servera ~]# systemctl reboot
```

4. Testez la configuration **autofs** avec les différents utilisateurs. Lorsque vous avez terminé, déconnectez-vous de **servera**.

- 4.1. Après le redémarrage de la machine **servera**, connectez-vous à **servera** en tant qu'utilisateur **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 4.2. Utilisez la commande **su - manager1** pour basculer vers l'utilisateur **manager1** et testez l'accès.

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

```
[student@servera ~]$ su - manager1
Password: redhat
[manager1@servera ~]$ ls -l /remote/management/
total 4
-rw-r--r--. 1 root managers 46 Apr  4 01:13 Welcome.txt
[manager1@servera ~]$ cat /remote/management>Welcome.txt
###Welcome to Management Folder on SERVERB###
[manager1@servera ~]$ echo TEST1 > /remote/management/Test.txt
[manager1@servera ~]$ cat /remote/management/Test.txt
TEST1
[manager1@servera ~]$ ls -l /remote/operation/
ls: cannot open directory '/remote/operation/': Permission denied
[manager1@servera ~]$ ls -l /remote/production/
ls: cannot open directory '/remote/production/': Permission denied
[manager1@servera ~]$ exit
logout
```

4.3. Basculez vers l'utilisateur dbuser1 et testez l'accès.

```
[student@servera ~]$ su - dbuser1
Password: redhat
[dbuser1@servera ~]$ ls -l /remote/production/
total 4
-rw-r--r--. 1 root production 46 Apr  4 01:13 Welcome.txt
[dbuser1@servera ~]$ cat /remote/production>Welcome.txt
###Welcome to Production Folder on SERVERB###
[dbuser1@servera ~]$ echo TEST2 > /remote/production/Test.txt
[dbuser1@servera ~]$ cat /remote/production/Test.txt
TEST2
[dbuser1@servera ~]$ ls -l /remote/operation/
ls: cannot open directory '/remote/operation/': Permission denied
[dbuser1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[dbuser1@servera ~]$ exit
logout
```

4.4. Basculez vers l'utilisateur contractor1 et testez l'accès.

```
[student@servera ~]$ su - contractor1
Password: redhat
[contractor1@servera ~]$ ls -l /remote/operation/
total 4
-rw-r--r--. 1 root operators 45 Apr  4 01:13 Welcome.txt
[contractor1@servera ~]$ cat /remote/operation>Welcome.txt
###Welcome to Operation Folder on SERVERB###
[contractor1@servera ~]$ echo TEST3 > /remote/operation/Test.txt
[contractor1@servera ~]$ cat /remote/operation/Test.txt
TEST3
[contractor1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[contractor1@servera ~]$ ls -l /remote/production/
```

**CHAPITRE 9 |** Accès au stockage rattaché au réseau

```
ls: cannot open directory '/remote/production/': Permission denied  
[contractor1@servera ~]$ exit  
logout
```

4.5. Explorez les options **mount** pour le partage NFS monté automatiquement.

```
[student@servera ~]$ mount | grep nfs  
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)  
serverb.lab.example.com:/shares/management on /remote/management type nfs4  
(rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,sync,proto=tcp,timeo=600,  
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)  
serverb.lab.example.com:/shares/operation on /remote/operation type nfs4  
(rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,sync,proto=tcp,timeo=600,  
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)  
serverb.lab.example.com:/shares/production on /remote/production type nfs4  
(rw,relatime,vers=4.2,rsize=262144,wsiz=262144,namlen=255,sync,proto=tcp,timeo=600,  
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)
```

4.6. Déconnectez-vous de servera.

```
[student@servera ~]$ exit  
logout
```

## Évaluation

À partir de workstation, exécutez la commande **lab netstorage-review grade** pour confirmer que vous avez réussi cet exercice.

```
[student@workstation ~]$ lab netstorage-review grade
```

## Fin

Sur workstation, exécutez la commande **lab netstorage-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netstorage-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris à effectuer les opérations suivantes :

- Monter et démonter une exportation NFS à partir de la ligne de commande.
- Configurer une exportation NFS de sorte qu'elle soit automatiquement montée au démarrage.
- Configurer le service de montage automatique avec des schémas de correspondance directe et indirecte et décrire leurs différences.
- Configurer les clients NFS afin qu'ils utilisent NFSv4 à l'aide du nouvel outil **nfsconf**.

## CHAPITRE 10

# Contrôle du processus de démarrage

### PROJET

Gérer le processus de démarrage pour contrôler les services proposés, et résoudre et corriger les problèmes.

### OBJECTIFS

- Décrire le processus de démarrage de Red Hat Enterprise Linux, définir la cible par défaut utilisée lors du démarrage et démarrer un système sur une cible différente de celle par défaut.
- Se connecter à un système et modifier le mot de passe root lorsque le mot de passe root actuel a été perdu.
- Réparer manuellement les problèmes de configuration ou de corruption du système de fichiers qui arrêtent le processus de démarrage.

### SECTIONS

- Sélection de la cible de démarrage (et exercice guidé)
- Réinitialisation du mot de passe root (et exercice guidé)
- Correction des problèmes de système de fichiers au démarrage (et exercice guidé)

### ATELIER

Contrôle du processus de démarrage

# SÉLECTION DE LA CIBLE DE DÉMARRAGE

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Décrire le processus de démarrage de Red Hat Enterprise Linux.
- Définir la cible par défaut utilisée lors du démarrage.
- Démarrer un système sur une cible non définie par défaut.

## DESCRIPTION DU PROCESSUS DE DÉMARRAGE DE RED HAT ENTERPRISE LINUX 8

Les ordinateurs actuels sont le fruit d'une collaboration complexe entre matériel et logiciel. Passer d'un état indéfini, hors tension, à un système opérationnel avec une invite de connexion nécessite l'interaction d'un grand nombre de composants matériels et logiciels. La liste suivante donne un aperçu rapide des tâches requises pour qu'un système **x86\_64** physique démarre sous Red Hat Enterprise Linux 8. La liste est sensiblement la même pour les machines virtuelles **x86\_64**, mais l'hyperviseur gère certaines des étapes spécifiques au matériel au niveau logiciel.

- La machine est mise sous tension. Le microprogramme du système, soit un UEFI moderne soit un BIOS plus ancien, exécute un test *Power On Self Test (POST)* et commence à initialiser une partie du matériel.

Elle est configurée à l'aide des écrans de configuration du BIOS ou de l'UEFI du système, auxquels vous accédez généralement par une combinaison de touches spécifique, par exemple **F2**, au tout début du processus de démarrage.

- Soit le microprogramme du système recherche un périphérique amorçable configuré dans le microprogramme de démarrage UEFI, soit il cherche un secteur *Master Boot Record (MBR)* sur tous les disques, dans l'ordre configuré dans le BIOS.

Il est configuré à l'aide des écrans de configuration du BIOS ou de l'UEFI du système, auxquels vous accédez généralement par une combinaison de touches spécifique, par exemple **F2**, au tout début du processus de démarrage.

- Le microprogramme du système lit un chargeur de démarrage depuis le disque, puis transmet le contrôle du système à ce chargeur. Sur un système Red Hat Enterprise Linux 8, le chargeur de démarrage est le *GRand Unified Bootloader version 2 (GRUB2)*.

Il est configuré à l'aide de la commande **grub2-install** qui installe GRUB2 en tant que chargeur de démarrage sur le disque.

- GRUB2 charge sa configuration à partir du fichier **/boot/grub2/grub.cfg** et affiche un menu dans lequel vous pouvez sélectionner le noyau à démarrer.

Il est configuré à l'aide du répertoire **/etc/grub.d/**, du fichier **/etc/default/grub** et de la commande **grub2-mkconfig** pour générer le fichier **/boot/grub2/grub.cfg**.

- Une fois le noyau sélectionné ou le délai expiré, le chargeur de démarrage charge le noyau et *initramfs* à partir du disque et les met en mémoire. Un **initramfs** est une archive qui contient

## CHAPITRE 10 | Contrôle du processus de démarrage

les modules de noyau pour tout le matériel nécessaire au démarrage, les scripts d'initialisation, etc. Sur Red Hat Enterprise Linux 8, **initramfs** contient un système entièrement opérationnel en soi.

Il est configuré à l'aide du répertoire **/etc/dracut.conf.d/**, de la commande **dracut** et de la commande **lsinitrd** pour examiner le fichier **initramfs**.

- Le chargeur de démarrage transmet le contrôle au noyau, en faisant passer les options éventuellement spécifiées sur la ligne de commande du noyau dans le chargeur de démarrage, et l'emplacement d'**initramfs** en mémoire.

Il est configuré à l'aide du répertoire **/etc/grub.d/**, du fichier **/etc/default/grub** et de la commande **grub2-mkconfig** pour générer le fichier **/boot/grub2/grub.cfg**.

- Le noyau initialise le matériel pour lequel il trouve un pilote dans **initramfs**, puis exécute le processus **/sbin/init** depuis **initramfs** en tant que PID 1. Sur Red Hat Enterprise Linux 8, **/sbin/init** est un lien vers **systemd**.

Il est configuré à l'aide du paramètre **init=** de la ligne de commande.

- L'instance de **systemd** qui se trouve sur **initramfs** exécute toutes les unités de la cible **initrd.target**. Cela inclut le montage du système de fichiers racine sur le répertoire **/sysroot**.

Elle est configurée en utilisant **/etc/fstab**.

- Le noyau fait basculer (pivoter) le système de fichiers racine de **initramfs** sur le système de fichiers root dans **/sysroot**. **systemd** se relance alors lui-même, à l'aide d'une copie de **systemd** installée sur le disque.
- **systemd** recherche une cible par défaut, entrée sur la ligne de commande du noyau ou configurée sur le système, puis démarre (et arrête) les unités en fonction de la configuration de cette cible, en résolvant automatiquement les dépendances entre les unités. Schématiquement, une cible **systemd** est un ensemble d'unités que le système doit activer pour atteindre l'état souhaité. Ces cibles démarrent généralement un écran de connexion texte ou graphique.

Il est configuré en utilisant **/etc/systemd/system/default.target** et **/etc/systemd/system/**.

## REDÉMARRAGE ET ARRÊT

Pour mettre un système hors tension ou le redémarrer à partir de la ligne de commande, vous pouvez utiliser la commande **systemctl**.

**systemctl poweroff** arrête tous les services en cours d'exécution, démonte tous les systèmes de fichiers (ou les remonte en lecture seule s'ils ne peuvent pas être démontés), puis arrête le système.

**systemctl reboot** arrête tous les services en cours d'exécution, démonte tous les systèmes de fichiers et redémarre le système.

Vous pouvez également utiliser la version abrégée de ces commandes, **poweroff** et **reboot**, qui sont des liens symboliques vers leurs équivalents **systemctl**.

**NOTE**

**systemctl halt** et **halt** peuvent aussi servir à arrêter le système, mais à la différence de **poweroff**, ces commandes ne coupent pas l'alimentation du système : elles l'amènent à un stade où l'on peut le mettre hors tension manuellement en toute sécurité.

## SÉLECTION D'UNE CIBLE SYSTEMD

Une cible systemd inclut un ensemble d'unités systemd que le système doit démarrer pour atteindre l'état voulu. Le tableau suivant liste les cibles les plus importantes.

### Cibles couramment utilisées

CIBLE	OBJET
graphical.target	Le système prend en charge plusieurs utilisateurs, ainsi que les connexions en mode texte et en mode graphique.
multi-user.target	Le système prend en charge plusieurs utilisateurs, et les connexions en mode texte uniquement.
rescue.target	Invite <b>sulogin</b> , initialisation du système de base terminée.
emergency.target	Invite <b>sulogin</b> , bascule complète d' <b>initramfs</b> et racine système montée sur / en lecture seule.

Une cible peut faire partie d'une autre cible. Par exemple, la cible graphical.target inclut la cible multi-user.target, qui dépend de la cible basic.target, etc. Vous pouvez afficher ces dépendances avec la commande suivante.

```
[user@host ~]$ systemctl list-dependencies graphical.target | grep target
graphical.target
* └─multi-user.target
*   ├─basic.target
*   | ├─paths.target
*   | ├─slices.target
*   | ├─sockets.target
*   | ├─sysinit.target
*   | | ├─cryptsetup.target
*   | | ├─local-fs.target
*   | | └─swap.target
...output omitted...
```

Pour lister les cibles disponibles, utilisez la commande suivante.

```
[user@host ~]$ systemctl list-units --type=target --all
UNIT           LOAD   ACTIVE   SUB    DESCRIPTION
-----          ----   -----   ---    -----
basic.target    loaded  active   active  Basic System
cryptsetup.target loaded  active   active  Local Encrypted Volumes
emergency.target loaded  inactive dead    Emergency Mode
getty-pre.target loaded  inactive dead    Login Prompts (Pre)
```

```
getty.target          loaded  active  active Login Prompts
graphical.target      loaded  inactive dead   Graphical Interface
...output omitted...
```

## Sélection d'une cible lors de l'exécution

Sur un système en cours d'exécution, les administrateurs peuvent basculer vers une autre cible en utilisant la commande **systemctl isolate**.

```
[root@host ~]# systemctl isolate multi-user.target
```

L'isolement d'une cible arrête tous les services qui ne sont pas requis par cette cible (ou par les dépendances associées), et lance au besoin tout service requis mais pas encore démarré.

Les cibles ne peuvent pas toutes être isolées. Vous ne pouvez isoler que les cibles qui ont **AllowIsolate=yes** défini dans leurs fichiers d'unité. Par exemple, vous pouvez isoler la cible graphique, mais pas la cible cryptsetup.

```
[user@host ~]$ systemctl cat graphical.target
# /usr/lib/systemd/system/graphical.target
...output omitted...
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes
[user@host ~]$ systemctl cat cryptsetup.target
# /usr/lib/systemd/system/cryptsetup.target
...output omitted...
[Unit]
Description=Local Encrypted Volumes
Documentation=man:systemd.special(7)
```

## Définition d'une cible par défaut

Quand le système démarre, systemd active la cible **default.target**. La cible par défaut dans **/etc/systemd/system/** doit logiquement correspondre à un lien symbolique vers **graphical.target** ou **multi-user.target**. Pour éviter la modification manuelle de ce lien symbolique, la commande **systemctl** propose deux sous-commandes pour le gérer : **get-default** et **set-default**.

```
[root@host ~]# systemctl get-default
multi-user.target
[root@host ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/
graphical.target.
[root@host ~]# systemctl get-default
graphical.target
```

## Sélection d'une cible différente au démarrage

Pour sélectionner une cible différente au démarrage, ajoutez l'option **systemd.unit=target.target** dans la ligne de commande du noyau, depuis le chargeur de démarrage.

Par exemple, pour démarrer le système dans un shell de secours où vous pouvez modifier la configuration du système avec presque aucun service en cours d'exécution, ajoutez l'option suivante à la ligne de commande du noyau à partir du chargeur de démarrage.

```
systemd.unit=rescue.target
```

Ce changement de configuration n'affecte qu'un seul démarrage, ce qui en fait un outil pratique pour résoudre les problèmes du processus de démarrage.

Pour sélectionner une autre cible via cette méthode, procédez comme suit :

1. Démarrez ou redémarrez le système.
2. Interrompez le compte à rebours du menu du chargeur de démarrage en appuyant sur une touche quelconque (à l'exception de la touche **Entrée** qui initierait un démarrage normal).
3. Placez le curseur sur l'entrée du noyau que vous voulez démarrer.
4. Appuyez sur **e** pour modifier l'entrée en cours.
5. Placez le curseur sur la ligne qui commence par **linux**. Il s'agit de la ligne de commande du noyau.
6. Ajoutez **systemd.unit=target.target**. Par exemple, **systemd.unit=emergency.target**.
7. Appuyez sur **Ctrl+x** pour démarrer le système en appliquant ces modifications.



### RÉFÉRENCES

**info grub2** (*manuel de GNU GRUB*)

Pages du manuel **bootup(7)**, **dracut.bootup(7)**, **lsinitrd(1)**,  
**systemd.target(5)**, **systemd.special(7)**, **sulogin(8)** et **systemctl(1)**

Pour plus d'informations, reportez-vous au chapitre *Managing services with systemd* du guide *Configuring basic system settings* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_basic\\_system\\_settings/#managing-services-with-systemd](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_basic_system_settings/#managing-services-with-systemd)

## ► EXERCICE GUIDÉ

# SÉLECTION DE LA CIBLE DE DÉMARRAGE

Dans cet exercice, vous allez déterminer la cible par défaut dans laquelle un système s'amorce et amorcer ce dernier dans d'autres cibles.

## RÉSULTATS

Vous devez pouvoir mettre à jour la cible système par défaut et utiliser une cible temporaire à partir du chargeur de démarrage.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab boot-selecting start**. Cette commande exécute un script de démarrage qui prépare workstation pour l'exercice.

```
[student@workstation ~]$ lab boot-selecting start
```

- ▶ 1. Sur workstation, ouvrez un terminal et vérifiez que la cible par défaut est graphical.target.

```
[student@workstation ~]$ systemctl get-default
graphical.target
```

- ▶ 2. Sur workstation, basculez manuellement vers la cible multi-user sans effectuer de redémarrage. Utilisez la commande **sudo** et lorsque vous y êtes invité, utilisez le mot de passe student.

```
[student@workstation ~]$ sudo systemctl isolate multi-user.target
[sudo] password for student: student
```

- ▶ 3. Accédez à la console basée sur du texte. Utilisez la séquence de touches **Ctrl+Alt+F1** à l'aide du bouton ou de l'entrée de menu correspondant. Connectez-vous en tant que root avec le mot de passe redhat.



### NOTE

Rappel : Si vous utilisez le terminal via une page Web, vous pouvez cliquer sur l'icône Afficher le clavier située sous la barre d'adresse de votre navigateur Web, puis à droite de l'adresse IP de la machine.

```
workstation login: root  
Password: redhat  
[root@workstation ~]#
```

- 4. Configurez **workstation** pour qu'il démarre automatiquement dans la cible **multi-user**, puis redémarrez **workstation** pour vérifier. Ensuite, redéfinissez la cible **systemd** par défaut sur **graphical**.

4.1. Utilisez la commande **systemctl set-default** pour définir la cible par défaut.

```
[root@workstation ~]# systemctl set-default multi-user.target  
Removed /etc/systemd/system/default.target.  
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/  
multi-user.target.
```

4.2. Redémarrez **workstation**.

```
[root@workstation ~]# systemctl reboot
```

Notez qu'après le redémarrage, le système présente une console basée sur le texte et non plus une connexion graphique.

4.3. Connectez-vous en tant que **root** avec le mot de passe **redhat**.

```
workstation login: root  
Password: redhat  
Last login: Thu Mar 28 14:50:53 on tty1  
[root@workstation ~]#
```

4.4. Redéfinissez la cible **systemd** par défaut sur **graphical**.

```
[root@workstation ~]# systemctl set-default graphical.target  
Removed /etc/systemd/system/default.target.  
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/  
graphical.target.
```

Ceci conclut la première partie de l'exercice dans laquelle vous vous êtes entraîné à définir la cible **systemd** par défaut.

- 5. Dans cette seconde partie de l'exercice, vous vous exercez à utiliser le mode de secours. Accédez au chargeur de démarrage en redémarrant **workstation** à nouveau. Depuis le menu du chargeur de démarrage, démarrez le système sur la cible **rescue**.

5.1. Lancez le redémarrage.

```
[root@workstation ~]# systemctl reboot
```

5.2. Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche pour interrompre le compte à rebours (à l'exception de la touche **Entrée** qui initierait un démarrage normal).

**CHAPITRE 10 |** Contrôle du processus de démarrage

- 5.3. Utilisez les touches de direction pour mettre en surbrillance la ligne du chargeur de démarrage par défaut.
- 5.4. Appuyez sur **e** pour modifier l'entrée en cours.
- 5.5. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**.
- 5.6. Appuyez sur **Fin** pour placer le curseur à la fin de la ligne.
- 5.7. Ajoutez **systemd.unit=rescue.target** à la fin de la ligne.
- 5.8. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.
- 5.9. Connectez-vous au mode de secours. Le mot de passe **root** est **redhat**. Vous devrez peut-être appuyer sur Entrée pour obtenir une invite claire.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@workstation ~]#
```

- 6. Vérifiez qu'en mode de secours le système de fichiers racine est en mode lecture/écriture.

```
[root@workstation ~]# mount
...output omitted...
/dev/vda1 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
...output omitted...
```

- 7. Appuyez sur **Ctrl+d** pour poursuivre le processus de démarrage.

Le système présente une connexion graphique. Connectez-vous en tant qu'utilisateur **student** avec le mot de passe **student**.

## Fin

Sur **workstation**, exécutez le script **lab boot-selecting finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab boot-selecting finish
```

L'exercice guidé est maintenant terminé.

# RÉINITIALISATION DU MOT DE PASSE ROOT

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir vous connecter à un système et modifier le mot de passe root lorsque le mot de passe root actuel a été perdu.

## RÉINITIALISATION DU MOT DE PASSE ROOT À PARTIR DU CHARGEUR DE DÉMARRAGE

La réinitialisation d'un mot de passe root perdu fait partie des tâches que tout administrateur système doit être capable d'effectuer. Si l'administrateur est toujours connecté, en tant qu'utilisateur sans privilège mais bénéficiant d'un accès **sudo** complet ou en tant que root, cette tâche est très simple. Par contre, si l'administrateur n'est pas connecté, cette tâche est légèrement plus complexe.

Plusieurs méthodes existent pour définir un nouveau mot de passe root. Un administrateur système peut par exemple démarrer le système à l'aide d'un Live CD, monter le système de fichiers racine à partir de là et modifier **/etc/shadow**. Dans cette section, nous expérimentons une méthode qui ne nécessite aucun recours à un support externe.



### NOTE

Sur Red Hat Enterprise Linux 6 et versions antérieures, les administrateurs peuvent démarrer le système au niveau d'exécution 1 pour obtenir une invite root. Sur une machine hébergeant Red Hat Enterprise Linux 8, les éléments les plus proches du niveau d'exécution 1 sont les cibles rescue et emergency qui utilisent le mot de passe root pour se connecter.

Sur Red Hat Enterprise Linux 8, il est possible de demander aux scripts qui sont exécutés à partir de **initramfs** de s'interrompre à certains moments, de fournir un shell root, puis de reprendre leur exécution lorsque ce shell se ferme. Cette opération est principalement utilisée à des fins de débogage. Cependant, elle peut également servir à récupérer un mot de passe root perdu.

Pour accéder à ce shell root, effectuez les étapes suivantes :

1. Redémarrez le système.
2. Interrompez le compte à rebours du chargeur de démarrage en appuyant sur une touche quelconque, à l'exception de la touche Entrée.
3. Placez le curseur sur l'entrée du noyau à démarrer.
4. Appuyez sur **e** pour modifier l'entrée sélectionnée.
5. Placez le curseur sur la ligne de commande du noyau (la ligne qui commence par **linux**).
6. Ajoutez **rd.break**. Avec cette option, le système s'arrête juste avant de transmettre le contrôle à partir du **initramfs** au système actuel.
7. Appuyez sur **Ctrl+x** pour démarrer le système en appliquant les modifications.

## CHAPITRE 10 | Contrôle du processus de démarrage

À ce stade, le système affiche un shell `root`, le système de fichiers racine réel du disque étant monté en lecture seule sur `/sysroot`. Étant donné que la résolution de problèmes implique souvent une modification du système de fichiers racine, vous devez modifier le système de fichiers racine en lecture et en écriture. L'étape suivante montre comment l'option `remount, rw` de la commande `mount` remonte le système de fichiers avec la nouvelle option (`rw`) définie.



### NOTE

Il se peut que des images préconstruites placent plusieurs arguments `console=` pour le noyau afin de prendre en charge un large éventail de scénarios de mise en œuvre. Ces arguments `console=` indiquent les périphériques à utiliser pour la sortie de la console. Il faut néanmoins faire attention avec `rd.break` car, bien que le système envoie les messages du noyau à toutes les consoles, l'invite finit par utiliser la dernière console, quelle qu'elle soit. Si vous n'obtenez pas l'invite, vous pouvez peut-être réorganiser temporairement les arguments `console=` lorsque vous éditez la ligne de commande du noyau à partir du chargeur de démarrage.



### IMPORTANT

Le système n'a pas encore activé SELinux. C'est pourquoi les fichiers que vous créez n'ont pas de contexte SELinux. Certains outils, comme la commande `passwd`, commencent par créer un fichier temporaire, puis le mettent à la place du fichier qu'ils sont censés modifier. Ainsi, ils créent un fichier sans contexte SELinux. Pour cette raison, lorsque vous utilisez la commande `passwd` avec `rd.break`, le fichier `/etc/shadow` n'obtient pas de contexte SELinux.

Pour réinitialiser le mot de passe `root` à ce stade, procédez comme suit :

1. Montez de nouveau `/sysroot` en lui associant un accès en lecture et en écriture.

```
switch_root:# mount -o remount,rw /sysroot
```

2. Accédez à une prison `chroot` dans laquelle `/sysroot` est considéré comme la racine de l'arborescence du système de fichiers.

```
switch_root:# chroot /sysroot
```

3. Définissez un nouveau mot de passe `root`.

```
sh-4.4# passwd root
```

4. Vérifiez que tous les fichiers non étiquetés, notamment `/etc/shadow` à ce stade, sont bien réétiquetés lors du démarrage.

```
sh-4.4# touch /.autorelabel
```

5. Saisissez deux fois `exit`. La première occurrence permet de quitter la prison `chroot` et la seconde, de quitter le shell de débogage du système `initramfs`.

À ce stade, le système poursuit son démarrage, effectue un réétiquetage complet de SELinux, puis redémarre.

## EXAMEN DES JOURNAUX

Il peut être utile de consulter les journaux des précédents démarrages qui ont échoué. Si les journaux système sont persistants lors des redémarrages, vous pouvez utiliser l'outil **journalctl** pour examiner ces journaux.

Souvenez-vous que, par défaut, les journaux du système sont conservés dans le répertoire **/run/log/journal**, ce qui signifie qu'ils sont effacés lorsque le système redémarre. Pour stocker les journaux dans le répertoire **/var/log/journal**, qui est conservé d'un redémarrage à un autre, définissez le paramètre **Storage** sur **persistent** dans **/etc/systemd/journald.conf**.

```
[root@host ~]# vim /etc/systemd/journald.conf
...output omitted...
[Journal]
Storage=persistent
...output omitted...
[root@host ~]# systemctl restart systemd-journald.service
```

Pour examiner les journaux d'un démarrage précédent, ajoutez l'option **-b** à **journalctl**. Sans aucun argument, l'option **-b** n'affiche que les messages depuis le dernier démarrage. Avec un nombre négatif en argument, elle affiche les journaux des démarrages précédents.

```
[root@host ~]# journalctl -b -1 -p err
```

Cette commande affiche l'ensemble des messages considérés au minimum comme des erreurs lors du démarrage précédent.

## CORRECTION DES PROBLÈMES DE DÉMARRAGE SYSTEMD

Pour résoudre les problèmes du service au démarrage, Red Hat Enterprise Linux 8 met à disposition les outils suivants.

### Activation du shell de débogage initial

Si vous activez le service **debug-shell** avec **systemctl enable debug-shell.service**, le système génère un shell root sur **TTY9 (Ctrl+Alt+F9)** au début de la séquence de démarrage. Ce shell est automatiquement connecté en tant que root. Ainsi, les administrateurs peuvent déboguer le système pendant que le système d'exploitation poursuit son démarrage.



#### MISE EN GARDE

N'oubliez pas de désactiver le service **debug-shell.service** lorsque vous avez terminé le débogage, car il laisse ouvert un shell root non authentifié, accessible à tout utilisateur disposant d'un accès à la console locale.

### Utilisation de cibles de secours et de réparation

Si vous ajoutez **systemd.unit=rescue.target** ou **systemd.unit=emergency.target** à la ligne de commande du noyau depuis le chargeur de démarrage, le système génère un shell de secours ou de réparation au lieu de démarrer normalement. Ces deux types de shell nécessitent un mot de passe **root**.

La cible `emergency` maintient le système de fichiers racine monté en lecture seule, alors que la cible `rescue` attend la fin de l'exécution de `sysinit.target`, pour qu'une plus grande partie du système soit initialisée, telle que le service de journalisation, ou les systèmes de fichiers. L'utilisateur root à ce stade ne peut pas modifier le répertoire `/etc/fstab` tant que le lecteur n'est pas remonté en lecture/écriture `mount -o remount,rw /`

Les administrateurs peuvent utiliser ces shells pour corriger tout problème empêchant le démarrage normal du système, par exemple une boucle de dépendance entre services ou un enregistrement incorrect dans `/etc/fstab`. Lorsque vous quittez ces shells, le processus d'amorçage normal se poursuit.

## Identification des tâches bloquées

Lors du démarrage, le programme `systemd` génère différentes tâches. Si certaines d'entre elles ne peuvent pas s'achever, elles bloquent l'exécution d'autres tâches. Pour examiner la liste des tâches en cours, les administrateurs peuvent utiliser la commande `systemctl list-jobs`. Toutes les tâches indiquées comme étant en cours d'exécution doivent se terminer pour que les tâches en attente puissent se poursuivre.



### RÉFÉRENCES

Pages du manuel `dracut cmdline(7)`, `systemd-journald(8)`,  
`journald.conf(5)`, `journalctl(1)` et `systemctl(1)`

## ► EXERCICE GUIDÉ

# RÉINITIALISATION DU MOT DE PASSE ROOT

Dans cet exercice, vous allez réinitialiser le mot de passe `root` sur un système.

## RÉSULTATS

Vous devez pouvoir réinitialiser un mot de passe `root` perdu.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab boot-resetting start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle réinitialise également le mot de passe `root` en une chaîne aléatoire et définit un délai d'expiration plus long pour le menu GRUB2.

```
[student@workstation ~]$ lab boot-resetting start
```

- ▶ 1. Redémarrez `servera` et interrompez le compte à rebours dans le menu du chargeur de démarrage.
  1. Localisez l'icône de la console `servera`, en fonction de votre environnement de formation. Ouvrez la console.  
Envoyez une commande **Ctrl+Alt+Suppr** à votre système en utilisant le bouton ou la commande de menu adéquat.
  2. Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche pour interrompre le compte à rebours, à l'exception de la touche **Entrée**.
- ▶ 2. Modifiez l'entrée du chargeur de démarrage par défaut, en mémoire, pour interrompre le processus de démarrage juste après que le noyau a monté l'ensemble des systèmes de fichiers, mais avant qu'il ne passe le contrôle au programme `systemd`.
  - 2.1. Utilisez les touches de direction pour mettre en surbrillance la ligne du chargeur de démarrage par défaut.
  - 2.2. Appuyez sur **e** pour modifier l'entrée en cours.
  - 2.3. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**.
  - 2.4. Appuyez sur **Fin** pour placer le curseur à la fin de la ligne.
  - 2.5. Ajoutez **rd.break** à la fin de la ligne.
  - 2.6. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.

- 3. À l'invite **switch\_root**, montez à nouveau le système de fichiers **/sysroot** en lecture et en écriture, puis utilisez **chroot** pour accéder à une prison **chroot** sur **/sysroot**.

```
switch_root:/# mount -o remount,rw /sysroot  
switch_root:/# chroot /sysroot
```

- 4. Rétablissez le mot de passe **root** à la valeur **redhat**.

```
sh-4.4# passwd root  
Changing password for user root.  
New password: redhat  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password: redhat  
passwd: all authentication tokens updated successfully.
```

- 5. Configurez le système pour qu'il effectue automatiquement un réétiquetage SELinux complet après le démarrage. Cette opération est nécessaire, car la commande **passwd** recrée le fichier **/etc/shadow** sans contexte SELinux.

```
sh-4.4# touch /.autorelabel
```

- 6. Saisissez **exit** deux fois pour poursuivre le démarrage normal du système. Le système exécute un réétiquetage SELinux, puis redémarre de lui-même. Lorsque le système est opérationnel, vérifiez votre travail en vous connectant en tant que **root** à la console. Utilisez **redhat** comme mot de passe.

## Fin

Sur **workstation**, exécutez le script **lab boot-resetting finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab boot-resetting finish
```

L'exercice guidé est maintenant terminé.

# CORRECTION DES PROBLÈMES DE SYSTÈME DE FICHIERS AU DÉMARRAGE

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réparer manuellement les problèmes de configuration ou de corruption du système de fichiers qui arrêtent le processus de démarrage.

## DIAGNOSTIC ET CORRECTION DES PROBLÈMES LIÉS AU SYSTÈME DE FICHIERS

Des erreurs dans **/etc/fstab** et des systèmes de fichiers corrompus peuvent empêcher le démarrage d'un système. Dans la plupart des cas, **systemd** lance un shell de réparation de secours qui nécessite le mot de passe **root**.

Le tableau suivant liste les erreurs les plus courantes et leur résultat.

### Problèmes liés aux systèmes de fichiers

PROBLÈME	RÉSULTAT
Système de fichiers corrompu	<b>systemd</b> tente de réparer le système de fichiers. Si le problème est trop grave pour une correction automatique, le système renvoie l'utilisateur à un shell de secours.
Périphérique ou UUID inexistant référencé dans <b>/etc/fstab</b>	Le programme <b>systemd</b> attend pendant une durée définie que le périphérique soit de nouveau disponible. Si le périphérique reste indisponible, le système renvoie l'utilisateur vers un shell de secours une fois le délai d'expiration passé.
Point de montage inexistant dans le fichier <b>/etc/fstab</b>	Le système renvoie l'utilisateur vers un shell de secours.
Option de montage incorrecte spécifiée dans <b>/etc/fstab</b>	Le système renvoie l'utilisateur vers un shell de secours.

Dans tous les cas, les administrateurs peuvent également utiliser la cible de secours pour diagnostiquer et corriger le problème, dans la mesure où aucun système de fichiers n'est monté avant l'affichage du shell de secours.



### NOTE

Lorsque vous utilisez le shell de secours pour résoudre les problèmes liés au système de fichiers, n'oubliez pas d'exécuter **systemctl daemon-reload** après avoir édité **/etc/fstab**. Sans ce rechargement, **systemd** peut continuer à utiliser l'ancienne version.

L'option **nofail** dans une entrée du fichier **/etc/fstab** permet au système de démarrer même si le montage de ce système de fichiers n'aboutit pas. N'utilisez pas cette option dans des

circonstances normales. Avec **nofail**, une application peut commencer sans son stockage, avec des conséquences graves.



## RÉFÉRENCES

Pages du manuel **systemd-fsck(8)**, **systemd-fstab-generator(8)** et **systemd.mount(5)**

## ► EXERCICE GUIDÉ

# CORRECTION DES PROBLÈMES DE SYSTÈME DE FICHIERS AU DÉMARRAGE

Dans cet exercice, vous allez récupérer un système à partir d'une configuration incorrecte dans **/etc/fstab** qui provoque l'échec du processus de démarrage.

## RÉSULTATS

Vous devez pouvoir diagnostiquer les problèmes liés à **/etc/fstab** et utiliser le mode de secours pour récupérer le système.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

Sur workstation, exécutez la commande **lab boot-repairing start**. Cette commande exécute un script de démarrage qui détermine si la machine servera est accessible sur le réseau. Elle introduit également un problème de système de fichiers, définit un délai d'expiration plus long pour le menu GRUB2 et redémarre servera.

```
[student@workstation ~]$ lab boot-repairing start
```

- ▶ 1. Accédez à la console servera et remarquez que le processus de démarrage est bloqué tôt.
  - 1.1. Localisez l'icône de la console servera, en fonction de votre environnement de formation. Ouvrez la console.  
Notez qu'un travail de démarrage semble ne pas vouloir se terminer. Prenez une minute pour réfléchir à la cause possible de ce problème.
  - 1.2. Pour redémarrer, envoyez une commande **Ctrl+Alt+Suppr** à votre système en utilisant le bouton ou l'entrée de menu adéquat. Avec ce problème de démarrage particulier, il est possible que cette séquence de touches n'interrompe pas immédiatement le travail en cours d'exécution et que vous deviez attendre son expiration avant le redémarrage du système.  
Si vous attendez que la tâche expire sans envoyer de message **Ctrl+Alt+Suppr**, le système génère finalement un shell de secours de lui-même.
  - 1.3. Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche pour interrompre le compte à rebours, à l'exception de la touche **Entrée**.
- ▶ 2. D'après l'erreur affichée lors du démarrage précédent, il semble qu'au moins certaines parties du système soient restées fonctionnelles. Comme vous connaissez le mot de passe root, redhat, vous pouvez tenter un démarrage de secours.
  - 2.1. Utilisez les touches de direction pour mettre en surbrillance la ligne du chargeur de démarrage par défaut.

**CHAPITRE 10 |** Contrôle du processus de démarrage

- 2.2. Appuyez sur **e** pour modifier l'entrée en cours.
- 2.3. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**.
- 2.4. Appuyez sur **Fin** pour placer le curseur à la fin de la ligne.
- 2.5. Ajoutez **systemd.unit=emergency.target** à la fin de la ligne.
- 2.6. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.

► **3.** Connectez-vous en mode de secours. Le mot de passe **root** est **redhat**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@servera ~]#
```

► **4.** Déterminez les systèmes de fichiers actuellement montés.

```
[root@servera ~]# mount
...output omitted...
/dev/vda1 on / type xfs (ro,relatime,seclabel,attr2,inode64,noquota)
...output omitted...
```

Notez que le shell du système de fichiers racine est monté en lecture seule.

► **5.** Remontez le système de fichiers racine en lecture et en écriture.

```
[root@servera ~]# mount -o remount,rw /
```

► **6.** Utilisez la commande **mount -a** pour essayer de monter l'ensemble des autres systèmes de fichiers. Avec l'option **--all (-a)**, la commande monte tous les systèmes de fichiers listés dans **/etc/fstab** qui ne sont pas encore montés.

```
[root@servera ~]# mount -a
mount: /RemoveMe: mount point does not exist.
```

► **7.** Éditez **/etc/fstab** pour corriger le problème.

- 7.1. Supprimez ou commentez la ligne incorrecte.

```
[root@servera ~]# vim /etc/fstab
...output omitted...
# /dev/sdz1  /RemoveMe  xfs  defaults  0 0
```

- 7.2. Mettez à jour **systemd** pour que le système enregistre la nouvelle configuration **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
[root@servera ~]#
```

► **8.** Vérifiez que le fichier **/etc/fstab** est maintenant correct en tentant de monter toutes les entrées.

```
[root@servera ~]# mount -a  
[root@servera ~]#
```

- 9. Redémarrez le système et attendez que le démarrage soit terminé. Le système doit maintenant démarrer normalement.

```
[root@servera ~]# systemctl reboot
```

## Fin

Sur workstation, exécutez le script **lab boot-repairing finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab boot-repairing finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# Contrôle du processus de démarrage

## Liste de contrôle des performances

Dans cet atelier, vous allez réinitialiser le mot de passe `root` sur un système, effectuer une récupération à partir d'une configuration incorrecte et définir la cible de démarrage par défaut.

## Résultats

Vous devez pouvoir réaliser les tâches suivantes :

- Réinitialiser un mot de passe `root` perdu.
- Diagnostiquer et corriger les problèmes de démarrage.
- Définir la cible `systemd` par défaut.

## Avant de commencer

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab boot-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle introduit également un problème de système de fichiers, redéfinit le mot de passe `root`, définit un délai d'expiration plus long pour le menu GRUB2 et redémarre `serverb`.

```
[student@workstation ~]$ lab boot-review start
```

1. Sur `serverb`, redéfinissez le mot de passe `root` sur `redhat`. Localisez l'icône de la console `serverb`, en fonction de votre environnement de formation. Travaillez depuis cette console.
2. Le système ne parvient pas à démarrer. Un travail de démarrage semble ne pas vouloir se terminer. À partir de la console, corrigez le problème.
3. Remplacez la cible `systemd` par défaut par `serverb` pour que le système démarre automatiquement une interface graphique au démarrage. Aucune interface graphique n'est encore installée sur `serverb`. Pour cet exercice, définissez uniquement la cible par défaut et n'installez pas les paquetages.

## Évaluation

À partir de `workstation`, exécutez le script `lab boot-review grade` pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab boot-review grade
```

## Fin

Sur workstation, exéutez le script **lab boot-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab boot-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# CONTRÔLE DU PROCESSUS DE DÉMARRAGE

## LISTE DE CONTRÔLE DES PERFORMANCES

Dans cet atelier, vous allez réinitialiser le mot de passe `root` sur un système, effectuer une récupération à partir d'une configuration incorrecte et définir la cible de démarrage par défaut.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Réinitialiser un mot de passe `root` perdu.
- Diagnostiquer et corriger les problèmes de démarrage.
- Définir la cible `systemd` par défaut.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab boot-review start`. Cette commande exécute un script de démarrage qui détermine si la machine `serverb` est accessible sur le réseau. Elle introduit également un problème de système de fichiers, redéfinit le mot de passe `root`, définit un délai d'expiration plus long pour le menu GRUB2 et redémarre `serverb`.

```
[student@workstation ~]$ lab boot-review start
```

1. Sur `serverb`, redéfinissez le mot de passe `root` sur `redhat`.

Localisez l'icône de la console `serverb`, en fonction de votre environnement de formation. Travaillez depuis cette console.

- 1.1. Envoyez une commande **`Ctrl+Alt+Suppr`** à votre système en utilisant le bouton ou l'entrée de menu adéquat.
- 1.2. Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche pour interrompre le compte à rebours, à l'exception de la touche **Entrée**.
- 1.3. Utilisez les touches de direction pour mettre en surbrillance la ligne du chargeur de démarrage par défaut.
- 1.4. Appuyez sur **e** pour modifier l'entrée en cours.
- 1.5. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**.
- 1.6. Appuyez sur **F10** pour placer le curseur à la fin de la ligne.

**CHAPITRE 10 |** Contrôle du processus de démarrage

- 1.7. Ajoutez **rd.break** à la fin de la ligne.
- 1.8. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.
- 1.9. À l'invite **switch\_root**, montez à nouveau le système de fichiers **/sysroot** en lecture et en écriture, puis utilisez **chroot** pour accéder à une prison **chroot** sur **/sysroot**.

```
switch_root:/# mount -o remount,rw /sysroot
switch_root:/# chroot /sysroot
```

- 1.10. Définissez le mot de passe **root** sur **redhat**.

```
sh-4.4# passwd root
Changing password for user root.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

- 1.11. Configurez le système pour qu'il effectue automatiquement un réétiquetage SELinux complet après le démarrage.

```
sh-4.4# touch /.autorelabel
```

- 1.12. Saisissez **exit** deux fois pour poursuivre le démarrage du système. Le système ne parvient pas à démarrer en raison d'un problème que vous corrigerez à l'étape suivante.
2. Le système ne parvient pas à démarrer. Un travail de démarrage semble ne pas vouloir se terminer. À partir de la console, corrigez le problème.
  - 2.1. Démarrez le système en mode de secours. Pour ce faire, redémarrez **serverb** en envoyant une commande **Ctrl+Alt+Suppr** à votre système en utilisant le bouton ou l'entrée de menu adéquat.
  - 2.2. Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche pour interrompre le compte à rebours, à l'exception de la touche **Entrée**.
  - 2.3. Utilisez les touches de direction pour mettre en surbrillance la ligne du chargeur de démarrage par défaut.
  - 2.4. Appuyez sur **e** pour modifier l'entrée en cours.
  - 2.5. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**.
  - 2.6. Appuyez sur **Fin** pour placer le curseur à la fin de la ligne.
  - 2.7. Ajoutez **systemd.unit=emergency.target** à la fin de la ligne.
  - 2.8. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.
  - 2.9. Connectez-vous en mode de secours. Le mot de passe **root** est **redhat**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@serverb ~]#
```

2.10. Remontez le système de fichiers / en lecture/écriture.

```
[root@serverb ~]# mount -o remount,rw /
```

2.11. Utilisez la commande **mount -a** pour essayer de monter l'ensemble des autres systèmes de fichiers.

```
[root@serverb ~]# mount -a
mount: /olddata: can't find UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc.
```

2.12. Modifiez **/etc/fstab** pour supprimer ou commenter la ligne incorrecte.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc  /olddata  xfs  defaults  0 0
```

2.13. Mettez à jour **systemd** pour que le système enregistre la nouvelle configuration **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]#
```

2.14. Vérifiez que le fichier **/etc/fstab** est maintenant correct en tentant de monter toutes les entrées.

```
[root@serverb ~]# mount -a
[root@serverb ~]#
```

2.15. Redémarrez le système et attendez que le démarrage soit terminé. Comme vous avez créé le fichier **/.autorelabel** lors de la première étape, après avoir défini le mot de passe **root**, le système exécute un réétiquetage SELinux, puis redémarre de lui-même. Le système doit maintenant démarrer normalement.

```
[root@serverb ~]# systemctl reboot
```

3. Remplacez la cible **systemd** par défaut par **serverb** pour que le système démarre automatiquement une interface graphique au démarrage.

Aucune interface graphique n'est encore installée sur **serverb**. Pour cet exercice, définissez uniquement la cible par défaut et n'installez pas les paquetages.

- 3.1. Connectez-vous à **serverb** en tant qu'utilisateur **root**. Utilisez **redhat** comme mot de passe.
- 3.2. Utilisez la commande **systemctl set-default** pour définir **graphical.target** comme cible par défaut.

```
[root@serverb ~]# systemctl set-default graphical.target
```

- 3.3. Utilisez la commande **systemctl get-default** pour vérifier votre travail.

```
[root@serverb ~]# systemctl get-default  
graphical.target
```

3.4. Déconnectez-vous de serverb.

```
[root@serverb ~]# exit
```

## Évaluation

À partir de workstation, exécutez le script **lab boot-review grade** pour confirmer que l'exercice est réussi.

```
[student@workstation ~]$ lab boot-review grade
```

## Fin

Sur workstation, exécutez le script **lab boot-review finish** pour mettre fin à l'atelier.

```
[student@workstation ~]$ lab boot-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- **systemctl reboot** et **systemctl poweroff** relancent et arrêtent le système, respectivement.
- **systemctl isolate target-name.target** passe à une nouvelle cible en cours d'exécution.
- **systemctl get-default** et **systemctl set-default** peuvent servir à invoquer la cible par défaut et à la définir.
- Utilisez **rd.break** sur la ligne de commande du noyau pour interrompre le processus de démarrage avant qu'**initramfs** ne rende la main. Le système de fichiers racine est monté en lecture seule sous **/sysroot**.
- La cible de secours peut également servir à diagnostiquer et corriger les problèmes de système de fichiers.



## CHAPITRE 11

# GESTION DE LA SÉCURITÉ RÉSEAU

### PROJET

Contrôler les connexions réseau aux services à l'aide du pare-feu du système et des règles SELinux.

### OBJECTIFS

- Accepter ou refuser les connexions réseau aux services système à l'aide des règles firewalld.
- Contrôler si les services réseau peuvent utiliser des ports réseau spécifiques en gérant les étiquettes de port SELinux.

### SECTIONS

- Gestion de pare-feu serveur (et exercice guidé)
- Contrôle de l'étiquetage de ports SELinux (et exercice guidé)

### ATELIER

Gestion de pare-feu serveur

# GESTION DE PARE-FEU SERVEUR

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir accepter ou refuser les connexions réseau aux services système à l'aide des règles firewalld.

## CONCEPTS D'ARCHITECTURE DE PARE-FEU

Le noyau Linux comprend **netfilter**, un framework pour les opérations de trafic réseau telles que le filtrage des paquets, la traduction d'adresses réseau et la traduction de ports. En mettant en œuvre des gestionnaires dans le noyau qui interceptent des appels de fonction et des messages, **netfilter** permet aux autres modules du noyau de s'interfacer directement avec la pile réseau du noyau. Le logiciel de pare-feu utilise ces scripts automatiques pour enregistrer les règles de filtrage et les fonctions de modification de paquets, permettant le traitement de chaque paquet passant par la pile réseau. Tout paquet réseau entrant, sortant ou transféré peut être inspecté, modifié, abandonné ou acheminé par programme avant d'atteindre les composants ou les applications de l'espace utilisateur. **Netfilter** est le composant principal des pare-feu dans Red Hat Enterprise Linux 8.

### Nftables améliore netfilter

Le noyau Linux comprend également **nftables**, un nouveau sous-système de filtrage et de classification des paquets qui comporte des portions améliorées du code **netfilter**, mais qui conserve l'architecture **netfilter** telle que les scripts automatiques de pile réseau, le système de suivi des connexions et la fonction de journalisation. Les avantages de la mise à jour **nftables** se traduisent par le traitement plus rapide des paquets, les mises à jour plus rapides des ensembles de règles et un traitement simultané IPv4 et IPv6 à partir des mêmes règles. Une autre différence majeure entre **nftables** et le framework original **netfilter** est leur interface. **Netfilter** est configuré via plusieurs frameworks d'utilitaires, notamment **iptables**, **ip6tables**, **arptables** et **eptables**, qui sont désormais obsolètes. **Nftables** se sert de l'utilitaire d'espace utilisateur **nft** unique. Avec ce dernier, toute la gestion des protocoles peut s'effectuer via une seule interface, éliminant ainsi les conflits historiques causés par divers systèmes frontaux et interfaces **netfilter**.

### Présentation de firewalld

**Firewalld** est un gestionnaire de pare-feu dynamique, une interface vers le framework **nftables** via la commande **nft**. Jusqu'à l'introduction de **nftables**, **firewalld** a utilisé la commande **iptables** pour configurer **netfilter** directement, en tant que solution améliorée du service **iptables**. Dans RHEL 8, **firewalld** reste le système frontal recommandé, gérant les ensembles de règles de pare-feu à l'aide de **nft**. **Firewalld** peut lire et gérer les fichiers de configuration et ensembles de règles **iptables**, en utilisant **xtables-nft-multi** pour traduire directement les objets **iptables** en règles et objets **nftables**. Bien que cette opération soit fortement déconseillée, **firewalld** peut être configuré pour redevenir le système principal **iptables** pour les cas d'utilisation complexes où les ensembles de règles **iptables** existants ne peuvent pas être correctement traités par les traductions **nft**.

Les applications interrogent le sous-système à l'aide de l'interface D-Bus. Le sous-système **firewalld**, disponible à partir du paquetage RPM **firewalld** n'est pas inclus dans une installation minimale, mais dans une installation de base. Avec **firewalld**, la gestion des pare-feu est

simplifiée en classifiant tout le trafic réseau en zones. Le trafic est dévié vers les règles de pare-feu de la zone appropriée, en fonction de critères tels que l'adresse IP source d'un paquet ou l'interface réseau du trafic entrant. Chaque zone peut avoir sa propre liste de ports et de services qui sont ouverts ou fermés.



### NOTE

Dans le cas des ordinateurs portables ou des autres machines qui changent régulièrement de réseau, on peut utiliser NetworkManager pour définir automatiquement la zone de pare-feu applicable à une connexion. Les zones sont personnalisables avec des règles adaptées à des connexions spécifiques.

Ceci se révèle particulièrement utile en cas d'alternance entre des réseaux sans fil personnels, professionnels et publics. Un utilisateur pourrait souhaiter que le service `sshd` de son système soit accessible lorsqu'il se connecte à son réseau domestique ou professionnel, mais pas s'il se connecte au réseau sans fil public du café local.

Firewalld vérifie l'adresse source de chaque paquet entrant dans le système. Si cette adresse source est liée à une zone spécifique, les règles de cette zone s'appliquent. Si l'adresse source n'est pas attribuée à une zone, `firewalld` associe le paquet à la zone de l'interface réseau entrante et les règles pour cette zone s'appliquent. Si, pour une raison quelconque, l'interface réseau n'est associée à aucune zone, alors `firewalld` associe le paquet à la zone par défaut.

La zone par défaut n'est pas une zone distincte, mais une désignation pour une zone existante. Initialement, `firewalld` désigne la zone `public` comme zone par défaut et met en correspondance l'interface loopback `lo` avec la zone `trusted`.

La plupart des zones autorisent le passage à travers le pare-feu du trafic qui correspond à une liste de ports et de protocoles spécifiques, tels que **631/udp**, ou à des services pré définis, tels que `ssh`. Si le trafic ne correspond pas à un port et un protocole ou service autorisé, il est généralement rejeté. (La zone `trusted`, qui autorise tout le trafic par défaut, est une exception à cette règle.)

## Zones prédéfinies

Firewalld comporte des zones prédéfinies que vous pouvez personnaliser. Par défaut, toutes les zones autorisent tout trafic entrant faisant partie intégrante d'une communication initiée par le système, ainsi que tout le trafic sortant. Le tableau suivant détaille cette configuration de zone initiale.

### Configuration par défaut des zones de firewalld

NOM DE LA ZONE	CONFIGURATION PAR DÉFAUT
<code>trusted</code>	Autorise tout le trafic entrant.
<code>accueil</code>	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond aux services pré définis <code>ssh</code> , <code>mdns</code> , <code>ipp-client</code> , <code>samba-client</code> ou <code>dhcpv6-client</code> .
<code>internal</code> ( <code>interne</code> )	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond aux services pré définis <code>ssh</code> , <code>mdns</code> , <code>ipp-client</code> , <code>samba-client</code> ou <code>dhcpv6-client</code> (comme la zone <code>home</code> , pour commencer).

NOM DE LA ZONE	CONFIGURATION PAR DÉFAUT
work	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond aux services prédéfinis ssh, ipp-client ou dhcpv6-client.
publique	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond aux services prédéfinis ssh ou dhcpv6-client. Zone par défaut des interfaces réseau nouvellement ajoutées.
external	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond au service prédéfini ssh. Le trafic IPv4 sortant transféré par l'intermédiaire de cette zone est camouflé pour donner l'impression qu'il provient de l'adresse IPv4 de l'interface réseau sortante.
dmz	Rejette le trafic entrant, sauf s'il est associé au trafic sortant ou s'il correspond au service prédéfini ssh.
bloc	Rejette la totalité du trafic entrant, sauf s'il est associé au trafic sortant.
drop	Ignore la totalité du trafic entrant, sauf s'il est associé au trafic sortant (sans même renvoyer de messages d'erreur ICMP).

Pour obtenir la liste des zones prédéfinies disponibles et de leur rôle, consultez **firewalld.zones(5)**.

## Services prédéfinis

Firewalld est également fourni avec plusieurs services prédéfinis. Ces définitions de service vous aident à identifier des services réseau particuliers à configurer. Au lieu d'avoir à rechercher des ports pertinents pour le service samba-client, spécifiez par exemple le service samba-client préconfiguré afin de définir les ports et les protocoles adéquats. Le tableau suivant liste les services prédéfinis utilisés dans la configuration initiale des zones de pare-feu.

### Sélection de services prédéfinis firewalld

NOM DU SERVICE	CONFIGURATION
ssh	Serveur SSH local. Trafic vers 22/tcp.
dhcpv6-client	Client DHCPv6 local. Trafic vers 546/udp sur le réseau IPv6 fe80::/64.
ipp-client	Impression IPP locale. Trafic vers le port 631/udp.
samba-client	Client de partage local de fichiers et d'imprimantes Windows. Trafic vers 137/udp et 138/udp.
mdns	Protocole de résolution de noms sur liaison locale appelée multidiffusion DNS (mDNS, Multicast DNS). Trafic vers 5353/udp pour les adresses de multidiffusion 224.0.0.251 (IPv4) ou ff02::fb (IPv6).

**NOTE**

De nombreux services prédéfinis sont inclus dans le paquetage `firewalld`. Utilisez `firewall-cmd --get-services` pour les lister. Les fichiers de configuration des services prédéfinis se trouvent dans `/usr/lib/firewalld/services`, dans un format défini par `firewalld.zone(5)`.

Utilisez les services prédéfinis ou spécifiez directement le port et le protocole requis. On peut aussi utiliser l'interface graphique de la console Web pour passer en revue les services prédéfinis et pour définir des services supplémentaires.

## CONFIGURATION DU PARE-FEU

Les administrateurs système peuvent interagir avec `firewalld` des trois manières suivantes :

- Modification directe des fichiers de configuration dans `/etc/firewalld/` (non décrit dans ce chapitre)
- Interface graphique de la console Web
- Outil de ligne de commande `firewall-cmd`

### Configuration des services de pare-feu à l'aide de la console Web

Pour configurer les services de pare-feu avec la console Web, connectez-vous à l'aide d'un accès avec privilège en cliquant sur l'option Reuse my password for privileged tasks. Cela permet à l'utilisateur d'exécuter des commandes, avec les priviléges sudo, afin de modifier les services de pare-feu.

Figure 11.1: Connexion avec privilège à la console Web

Cliquez sur l'option Networking dans le menu de navigation de gauche pour afficher la section Firewall dans la page principale de mise en réseau. Cliquez sur le lien Firewall pour accéder à la liste des services autorisés.

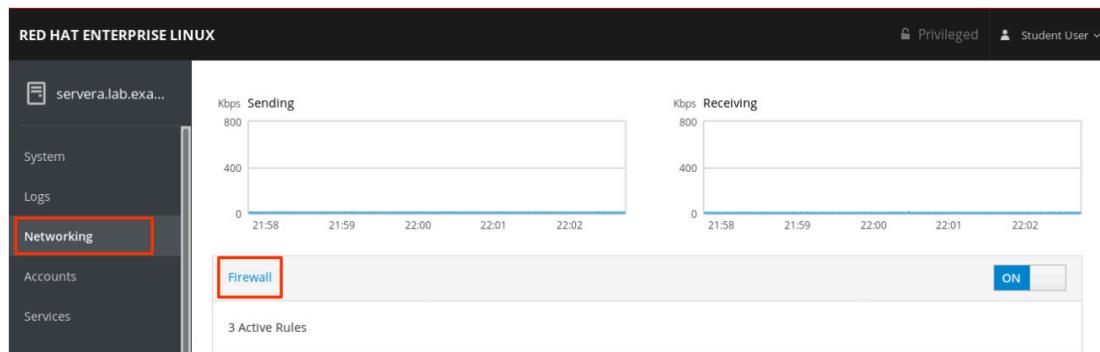


Figure 11.2: Mise en réseau de la console Web

Les services autorisés listés sont ceux qui sont actuellement autorisés par le pare-feu. Cliquez sur la flèche (>) à gauche du nom du service pour afficher les détails du service. Pour ajouter un service, cliquez sur le bouton Add Services... dans le coin supérieur droit de la page Firewall Allowed Services.

The screenshot shows the 'Networking > Firewall' section of the web interface. A table lists 'Allowed Services' with columns for 'Service', 'TCP', and 'UDP'. The 'SSH' row is selected, indicated by a blue highlight. The 'Add Services...' button in the top right corner is also highlighted with a red box. On the left sidebar, the 'Services' option is selected and highlighted with a red box.

Service	TCP	UDP
Cockpit	9090	
DHCPv6 Client		546
SSH	22	

Figure 11.3: Liste des services autorisés par le pare-feu de la console Web

La page Add Services affiche les services prédéfinis disponibles.

The screenshot shows the 'Add Services' dialog box. At the top, the title 'Add Services' is highlighted with a red box. Below it is a 'Filter Services' input field containing the text 'http'. A list of services follows, each with a checkbox. The services listed are: Red Hat Satellite 6, Amanda Backup Client, Amanda Backup Client (kerberized), amqp, amqps, and apcupsd. At the bottom right of the dialog are 'Cancel' and 'Add Services' buttons, with 'Add Services' highlighted with a red box.

Figure 11.4: Ajout d'une interface de services par la console Web

Pour sélectionner un service, faites défiler la liste ou entrez une sélection dans la zone de texte Filter Services. Dans l'exemple suivant, la chaîne **http** est saisie dans la zone de texte de

recherche pour trouver les services contenant cette chaîne, c'est-à-dire les services liés au Web. Cochez la case située à gauche des services pour autoriser le passage à travers du pare-feu. Cliquez sur le bouton Add Services pour terminer le processus.

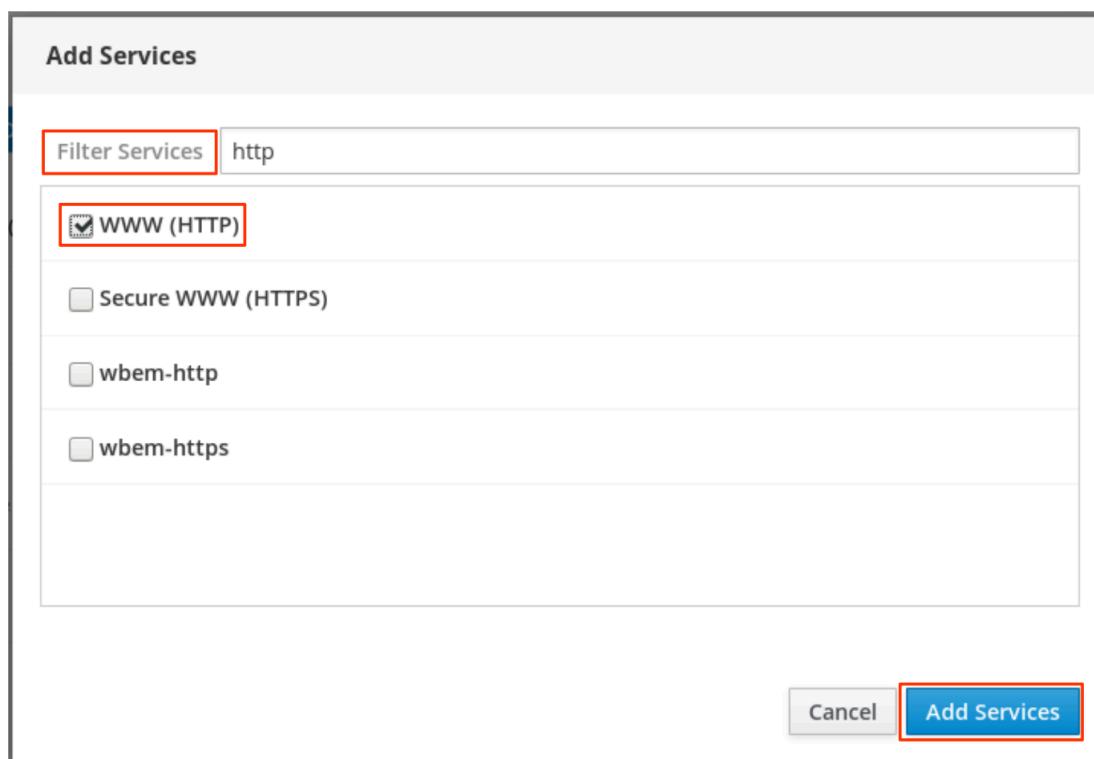


Figure 11.5: Recherche par filtre des services de la console Web

L'interface revient à la page Firewall Allowed Services dans laquelle vous pouvez consulter la liste mise à jour des services autorisés.

Service	TCP	UDP
Cockpit	9090	
DHCPv6 Client		546
SSH	22	
WWW (HTTP)	80	

Figure 11.6: Liste des services de la console Web

## Configuration du pare-feu à partir de la ligne de commande

La commande **firewall-cmd** interagit avec le gestionnaire de pare-feu dynamique **firewalld**. Il est installé dans le cadre du paquetage *firewalld* principale. Il s'adresse aux administrateurs

qui préfèrent agir à partir de la ligne de commande, pour travailler sur des systèmes dépourvus d'environnement graphique ou pour écrire un script de configuration de pare-feu.

Le tableau ci-après liste plusieurs commandes **firewall-cmd** fréquemment utilisées avec leur description. Notez que, sauf indication contraire, quasiment toutes les commandes fonctionnent avec la configuration *runtime*, à moins que l'option **--permanent** n'ait été spécifiée. Si l'option **--permanent** est spécifiée, vous devez activer le paramètre en exécutant également la commande **firewall-cmd --reload**, qui lit la configuration permanente actuelle et l'applique comme nouvelle configuration d'exécution. De nombreuses commandes listées utilisent l'option **--zone=ZONE** pour déterminer la zone sur laquelle elles portent. Lorsqu'un masque de réseau est requis, utilisez la notation CIDR, telle que 192.168.1/24.

COMMANDES FIREWALL-CMD	EXPLICATION
<b>--get-default-zone</b>	Indique l'actuelle zone par défaut.
<b>--set-default-zone=ZONE</b>	Définit la zone par défaut. Cette valeur modifie la configuration en cours d'exécution et la configuration permanente.
<b>--get-zones</b>	Répertorie toutes les zones disponibles.
<b>--get-active-zones</b>	Liste toutes les zones en cours d'utilisation (auxquelles sont associées une interface ou une source), ainsi que les informations sur leur interface et leur source.
<b>--add-source=CIDR [ --zone=ZONE]</b>	Achemine l'ensemble du trafic provenant de l'adresse IP ou du réseau/masque de réseau vers la zone spécifiée. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--remove-source=CIDR [ --zone=ZONE]</b>	Supprime la règle qui achemine, à partir de la zone, l'ensemble du trafic provenant de l'adresse IP ou du réseau/masque de réseau. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--add-interface=INTERFACE [ --zone=ZONE]</b>	Achemine l'ensemble du trafic provenant de l' <b>INTERFACE</b> vers la zone spécifiée. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--change-interface=INTERFACE [ --zone=ZONE]</b>	Associe l'interface à la <b>ZONE</b> plutôt qu'à sa zone actuelle. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--list-all [ --zone=ZONE]</b>	Liste l'ensemble des interfaces, sources, services et ports configurés pour la <b>ZONE</b> . Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.

COMMANDES FIREWALL-CMD	EXPLICATION
<b>--list-all-zones</b>	Récupère toutes les informations relatives à toutes les zones (interfaces, sources, ports, services).
<b>--add-service=SERVICE [ --zone=ZONE]</b>	Autorise le trafic vers le <i>SERVICE</i> . Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--add-port=PORT/PROTOCOL [ --zone=ZONE]</b>	Autorise le trafic vers le ou les ports <i>PORT/PROTOCOL</i> . Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--remove-service=SERVICE [ --zone=ZONE]</b>	Supprime le <i>SERVICE</i> de la liste autorisée pour la zone. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--remove-port=PORT/PROTOCOL [ --zone=ZONE]</b>	Supprime les ports <i>PORT/PROTOCOL</i> de la liste autorisée pour la zone. Si aucune option <b>--zone=</b> n'est spécifiée, le programme utilise la zone par défaut.
<b>--reload</b>	Abandonne la configuration en cours d'exécution et applique la configuration permanente.

Dans les exemples ci-après, la zone par défaut est définie sur `dmz`, l'ensemble du trafic provenant du réseau `192.168.0.0/24` est affecté à la zone `internal` et les ports réseau pour le service `mysql` sont ouverts dans la zone `internal`.

```
[root@host ~]# firewall-cmd --set-default-zone=dmz
[root@host ~]# firewall-cmd --permanent --zone=internal \
--add-source=192.168.0.0/24
[root@host ~]# firewall-cmd --permanent --zone=internal --add-service=mysql
[root@host ~]# firewall-cmd --reload
```



### NOTE

Dans les cas où la syntaxe de base de `firewall-cmd` s'avère trop limitée, vous pouvez également ajouter des *règles riches* (« rich-rules »), une syntaxe plus expressive pour écrire des règles plus complexes. Enfin, si la syntaxe de règles riches ne suffit pas, vous pouvez également recourir aux règles de *configuration directe*, qui correspondent à une syntaxe `nft` brute combinée à des règles `firewall-cmd`.

Ces modes avancés ne seront pas abordés dans le présent chapitre.



### RÉFÉRENCES

Pages du manuel `firewall-cmd(1)`, `firewall(1)`, `firewalld.zone(5)`, `firewalld.zones(5)` et `nft(8)`

## ► EXERCICE GUIDÉ

# GESTION DE PARE-FEU SERVEUR

Dans cet exercice, vous contrôlerez l'accès aux services système en ajustant les règles de pare-feu du système avec `firewalld`.

## RÉSULTATS

Vous devez pouvoir configurer des règles de pare-feu pour contrôler l'accès aux services.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

À partir de `workstation`, exécutez la commande **`lab netsecurity-firewalls start`**. La commande exécute un script de démarrage pour déterminer si l'hôte `servera` est accessible sur le réseau.

```
[student@workstation ~]$ lab netsecurity-firewalls start
```

- ▶ 1. À partir de `workstation`, connectez-vous via SSH à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- ▶ 2. Sur le système `servera`, assurez-vous que les paquetages `httpd` et `mod_ssl` sont tous deux installés. Ces paquetages fournissent le serveur Web Apache que vous allez protéger avec un pare-feu, ainsi que les extensions nécessaires pour permettre au serveur Web de publier du contenu via SSL.

```
[student@servera ~]$ sudo yum install httpd mod_ssl
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- ▶ 3. En tant qu'utilisateur `student` sur `servera`, créez le fichier `/var/www/html/index.html`. Ajoutez la ligne de texte : **I am servera**.

```
[student@servera ~]$ sudo bash -c \
"echo 'I am servera.' > /var/www/html/index.html"
```

- 4. Démarrez et activez le service `httpd` sur votre système `servera`.

```
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/
lib/systemd/system/httpd.service.
```

- 5. Quittez `servera`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

- 6. À partir de `workstation`, tentez d'accéder à votre serveur Web sur `servera` en utilisant à la fois le port en texte clair 80/TCP et le port SSL encapsulé 443/TCP. Les deux tentatives doivent échouer.

- 6.1. La commande ci-après doit échouer :

```
[student@workstation ~]$ curl -k http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```

- 6.2. La commande ci-après doit également échouer :

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 443: No route to host
```

- 7. Connectez-vous à `servera` en tant qu'utilisateur `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 8. Sur `servera`, assurez-vous que le service `nftables` est masqué et le service `firewalld` est activé et en cours d'exécution.

- 8.1. Déterminez si l'état du service `nftables` est masked.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student: student
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor
  preset: disabled)
  Active: inactive (dead)
    Docs: man:nft(8)
```

Les résultats montrent que `nftables` est désactivé et inactif mais non masqué. Exécutez la commande suivante pour masquer le service.

**CHAPITRE 11 |** Gestion de la sécurité réseau

```
[student@servera ~]$ sudo systemctl mask nftables  
Created symlink /etc/systemd/system/nftables.service → /dev/null.
```

8.2. Vérifiez si l'état du service `nftables` est masked.

```
[student@servera ~]$ sudo systemctl status nftables  
● nftables.service  
  Loaded: masked (Reason: Unit nftables.service is masked.)  
  Active: inactive (dead)
```

8.3. Vérifiez que l'état du service `firewalld` est activé et en cours d'exécution.

```
[student@servera ~]$ sudo systemctl status firewalld  
● firewalld.service - firewalld - dynamic firewall daemon  
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor  
  preset: enabled)  
  Active: active (running) since Wed 2019-05-22 15:36:02 CDT; 5min ago  
    Docs: man:firewalld(1)  
  Main PID: 703 (firewalld)  
    Tasks: 2 (limit: 11405)  
   Memory: 29.8M  
     CGroup: /system.slice/firewalld.service  
             └─703 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --  
       nopid  
  
May 22 15:36:01 jegui.ilt.example.com systemd[1]: Starting firewalld - dynamic  
firewall daemon...  
May 22 15:36:02 jegui.ilt.example.com systemd[1]: Started firewalld - dynamic  
firewall daemon.
```

8.4. Quittez servera.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

► **9.** À partir de `workstation`, ouvrez Firefox et connectez-vous à la console Web exécutée sur `servera` pour ajouter le service `httpd` à la zone de réseau public.

9.1. Ouvrez Firefox et accédez à `https://servera.lab.example.com:9090` pour ouvrir la console Web. Acceptez le certificat autosigné par `servera` en ajoutant une exception.

9.2. Cochez la case en regard de `Reuse my password for privileged tasks` pour disposer des priviléges d'administration.

Connectez-vous en tant qu'utilisateur `student` avec le mot de passe `student`.

9.3. Cliquez sur `Networking` dans la barre de navigation de gauche.

9.4. Cliquez sur le lien `Firewall` dans la page `Networking` principale.

- 9.5. Cliquez sur le bouton Add Services... situé dans le coin supérieur droit de la page Firewall.
  - 9.6. Dans l'interface utilisateur Add Services, faites défiler ou utilisez Filter Services pour localiser et sélectionner la case à cocher en regard du service Secure WWW (HTTPS).
  - 9.7. Cliquez sur le bouton Add Services situé dans le coin inférieur droit de l'interface utilisateur Add Services.
- ▶ 10. Revenez à un terminal sur **workstation** et vérifiez votre travail en essayant d'afficher le contenu du serveur Web de **servera**.

10.1. La commande ci-après doit échouer :

```
[student@workstation ~]$ curl -k http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```

10.2. La commande suivante doit aboutir :

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
I am servera.
```



#### NOTE

Si vous utilisez Firefox pour vous connecter au serveur Web, une invite de vérification du certificat de l'hôte s'affichera si le pare-feu a été franchi avec succès.

## Terminer

Sur **workstation**, exécutez le script **lab netsecurity-firewalls finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netsecurity-firewalls finish
```

L'exercice guidé est maintenant terminé.

# CONTRÔLE DE L'ÉTIQUETAGE DE PORTS SELINUX

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir vérifier que les ports réseau ont le type SELinux approprié permettant aux services de s'y lier.

## ÉTIQUETAGE DES PORTS SELINUX

L'action de SELinux ne se limite pas simplement à l'étiquetage des fichiers et des processus. La politique SELinux assure également l'application stricte du trafic sur le réseau. L'une des méthodes utilisées par SELinux pour contrôler le trafic consiste à étiqueter les ports réseau. Par exemple, dans la politique **targeted**, le port **22/TCP** présente l'étiquette **ssh\_port\_t** correspondante. L'étiquette **http\_port\_t** est associée aux ports HTTP par défaut, **80/TCP** et **443/TCP**.

Lorsqu'un processus veut écouter un port, SELinux vérifie que l'étiquette correspondante à ce processus (le domaine) autorise la liaison à ce port. Cette mesure permet d'éviter qu'un service non autorisé n'accapare les ports utilisés par d'autres services réseau (légitimes).

## GESTION DE L'ÉTIQUETAGE DES PORTS SELINUX

Si vous décidez d'exécuter un service sur un port non standard, SELinux bloquera presque certainement le trafic. Dans ce cas, vous devez mettre à jour les étiquettes de port SELinux. Dans certains cas, la politique **targeted** a déjà étiqueté le port avec un type utilisable. Par exemple, comme le port **8008/TCP** est souvent utilisé pour les applications Web, il présente déjà l'étiquette **http\_port\_t**, le type de port par défaut du serveur Web.

### Affichage des étiquettes de port

Pour obtenir un aperçu de toutes les étiquettes de port assignées, exécutez la commande **semanage port -l**. L'option **-l** liste toutes les affectations actuelles, au format suivant :

```
port_label_t      tcp|udp      comma-separated, list, of, ports
```

Exemple de sortie :

```
[root@host ~]# semanage port -l
...output omitted...
http_cache_port_t      tcp    8080, 8118, 8123, 10001-10010
http_cache_port_t      udp    3130
http_port_t             tcp    80, 81, 443, 488, 8008, 8009, 8443, 9000
...output omitted...
```

Pour affiner la recherche, utilisez la commande **grep** :

```
[root@host ~]# semanage port -l | grep ftp
ftp_data_port_t          tcp      20
ftp_port_t                tcp      21, 989, 990
ftp_port_t                udp      989, 990
tftp_port_t               udp      69
```

Notez qu'une étiquette de port peut apparaître deux fois dans une sortie, une fois pour TCP et une fois pour UDP.

## Gestion des étiquettes de port

Utilisez la commande **semanage** pour assigner de nouvelles étiquettes de port et supprimer ou modifier les étiquettes de port existantes.



### IMPORTANT

La plupart des services standard disponibles dans la distribution Linux fournissent un module de politique SELinux qui définit les étiquettes sur les ports. Vous ne pouvez pas changer les étiquettes sur ces ports en utilisant **semanage** ; pour ce faire, vous devez remplacer le module de politique. L'écriture et la génération des modules de politique ne sont pas abordées dans ce cours.

Pour ajouter un port à une étiquette existante (type), utilisez la syntaxe suivante. L'option **-a** ajoute une nouvelle étiquette de port, **-t** désigne le type et **-p** le protocole.

```
[root@host ~]# semanage port -a -t port_label -p tcp|udp PORTNUMBER
```

Par exemple, pour autoriser un service gopher à écouter un port **71/TCP** :

```
[root@host ~]# semanage port -a -t gopher_port_t -p tcp 71
```

Pour afficher les modifications locales de la politique par défaut, les administrateurs peuvent ajouter l'option **-C** à la commande **semanage**.

```
[root@host ~]# semanage port -l -C
SELinux Port Type          Proto    Port Number
gopher_port_t               tcp      71
```



### NOTE

La politique **targeted** comprend un grand nombre de types de port.

Les pages de manuel spécifiques aux services SELinux du paquetage *selinux-policy-doc* comprennent de la documentation relative aux types SELinux, aux valeurs booléennes et aux types de port. Si ces pages ne sont pas encore installées sur votre système, suivez cette procédure :

```
[root@host ~]# yum -y install selinux-policy-doc
[root@host ~]# man -k _selinux
```

## Suppression des étiquettes de port

La syntaxe permettant de supprimer une étiquette de port personnalisée est identique à celle qui permet d'en ajouter une, mais au lieu d'utiliser l'option **-a** (Add), utilisez l'option **-d** (Delete).

Par exemple, pour supprimer la liaison du port **71/TCP** à **gopher\_port\_t** :

```
[root@host ~]# semanage port -d -t gopher_port_t -p tcp 71
```

## Modification des liaisons de port

Pour modifier une liaison de port, peut-être parce que les exigences ont changé, utilisez l'option **-m** (Modify). Ce processus est plus efficace que de supprimer l'ancienne liaison pour en ajouter une nouvelle.

Par exemple, pour remplacer l'étiquette du port **71/TCP gopher\_port\_t** par **http\_port\_t**, l'administrateur peut utiliser la commande suivante :

```
[root@server ~]# semanage port -m -t http_port_t -p tcp 71
```

Comme auparavant, affichez la modification à l'aide de la commande **semanage**.

```
[root@server ~]# semanage port -l -c
SELinux Port Type          Proto    Port Number

http_port_t                tcp      71
[root@server ~]# semanage port -l | grep http
http_cache_port_t           tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t           udp      3130
http_port_t                tcp      71, 80, 81, 443, 488, 8008, 8009, 8443,
                                9000
pegasus_http_port_t         tcp      5988
pegasus_https_port_t        tcp      5989
```



### RÉFÉRENCES

Pages de manuel **semanage(8)**, **semanage-port(8)** et **\*\_selinux(8)**

## ► EXERCICE GUIDÉ

# CONTROLE DE L'ÉTIQUETAGE DE PORTS SELINUX

Au cours de cet atelier, vous allez configurer votre système `servera` de sorte qu'il autorise l'accès HTTP sur un port non standard.

## RÉSULTATS :

Vous allez configurer un serveur Web exécuté sur `servera` pour distribuer du contenu à l'aide d'un port non standard.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur `student` à `workstation` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab netsecurity-ports start`. Cette commande exécute un script de démarrage qui détermine si la machine `servera` est accessible sur le réseau. Elle installe également le service `httpd` et configure le pare-feu sur `servera` afin d'autoriser les connexions http.

```
[student@workstation ~]$ lab netsecurity-ports start
```

Votre entreprise est en train de déployer une nouvelle application Web personnalisée. Malheureusement, l'application Web est exécutée sur un port non standard, dans ce cas le port **82/TCP**.

Un de vos administrateurs débutants a déjà configuré l'application sur votre `servera`. Cependant, le contenu du serveur Web n'est pas accessible.

- 1. Utilisez la commande `ssh` pour vous connecter à `servera` en tant qu'utilisateur `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Utilisez la commande `sudo -i` pour basculer vers l'utilisateur `root`. Le mot de passe de l'utilisateur `student` est `student`.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Essayez de résoudre le problème de contenu Web en redémarrant le service `httpd`.

- 3.1. Utilisez la commande **systemctl** pour redémarrer **httpd.service**. Cette commande doit échouer.

```
[root@servera ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 3.2. Utilisez la commande **systemctl status -l** pour afficher l'état du service httpd. Remarquez le message d'erreur **permission denied**.

```
[root@servera ~]# systemctl status -l httpd.service
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Active: failed (Result: exit-code) since Mon 2019-04-08 14:23:29 CEST; 3min 33s ago
    Docs: man:httpd.service(8)
   Process: 28078 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
   Main PID: 28078 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."

Apr 08 14:23:29 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission denied:
AH00072: make_sock: could not bind to address [::]:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission denied:
AH00072: make_sock: could not bind to address 0.0.0.0:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: no listening sockets available, shutting down
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: AH00015: Unable to open logs
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Apr 08 14:23:29 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

- 3.3. Utilisez la commande **sealert** pour vérifier si SELinux empêche **httpd** de se lier au port **82/TCP**.

```
[root@servera ~]# sudo sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----
SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 82.

***** Plugin bind_ports (99.5 confidence) suggests *****

If you want to allow /usr/sbin/httpd to bind to network port 82
Then you need to modify the port type.
```

```
Do  
# semanage port -a -t PORT_TYPE -p tcp 82  
    where PORT_TYPE is one of the following: http_cache_port_t, http_port_t,  
    jboss_management_port_t, jboss.messaging_port_t, ntop_port_t, puppet_port_t.  
...output omitted...  
Raw Audit Messages  
type=AVC msg=audit(1554726569.188:852): avc: denied { name_bind } for  
    pid=28393 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0  
    tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0  
...output omitted...
```

- ▶ 4. Configurez SELinux de sorte qu'il autorise **httpd** à se lier au port **82/TCP**, puis redémarrez le service **httpd.service**.

- 4.1. Utilisez la commande **semanage** pour trouver un type de port approprié pour le port **82/TCP**.

```
[root@servera ~]# semanage port -l | grep http  
http_cache_port_t          tcp      8080, 8118, 8123, 10001-10010  
http_cache_port_t          udp      3130  
http_port_t                tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000  
pegasus_http_port_t        tcp      5988  
pegasus_https_port_t       tcp      5989
```

**http\_port\_t** contient les ports HTTP par défaut, **80/TCP** et **443/TCP**. Il s'agit du type de port correct pour le serveur Web.

- 4.2. À l'aide de la commande **semanage**, attribuez le type **http\_port\_t** au port **82/TCP**.

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
```

- 4.3. Utilisez la commande **systemctl** pour redémarrer le service **httpd.service**. La commande suivante doit aboutir.

```
[root@servera ~]# systemctl restart httpd.service
```

- ▶ 5. Vérifiez si vous pouvez maintenant accéder au serveur Web fonctionnant sur le port **82/TCP**. Utilisez la commande **curl** pour accéder au service Web à partir de **servera**.

```
[root@servera ~]# curl http://servera.lab.example.com:82  
Hello
```

- ▶ 6. Dans une autre fenêtre de terminal, vérifiez si vous pouvez accéder au nouveau service Web à partir de **workstation**. Utilisez la commande **curl** pour accéder au service Web à partir de **workstation**.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82  
curl: (7) Failed to connect to servera.example.com:82; No route to host
```

Cette erreur signifie que vous ne pouvez toujours pas vous connecter au service Web à partir de **workstation**.

- 7. Sur servera, ouvrez le port **82/TCP** sur le pare-feu.
- 7.1. Utilisez la commande **firewall-cmd** pour ouvrir le port **82/TCP** dans la configuration permanente de la zone par défaut du pare-feu sur servera.

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp  
success
```

7.2. Appliquez les modifications de votre pare-feu sur servera.

```
[root@servera ~]# firewall-cmd --reload  
success
```

- 8. Utilisez la commande **curl** pour accéder au service Web à partir de workstation.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82  
Hello
```

- 9. Quittez servera.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

## Terminer

Sur workstation, exécutez le script **lab netsecurity-ports finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netsecurity-ports finish
```

L'exercice guidé est maintenant terminé.

## ► OPEN LAB

# GESTION DE LA SÉCURITÉ RÉSEAU

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez configurer les paramètres de pare-feu et SELinux pour permettre l'accès à plusieurs serveurs Web exécutés sur serverb.

## RÉSULTATS

Vous devez pouvoir configurer les paramètres de pare-feu et SELinux sur un hôte de serveur Web.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

À partir de workstation, exécutez la commande **lab netsecurity-review start**. La commande exécute un script de démarrage pour déterminer si l'hôte serverb est accessible sur le réseau.

```
[student@workstation ~]$ lab netsecurity-review start
```

Votre société a décidé d'exécuter une nouvelle application Web. Celle-ci écoute sur les ports **80/TCP** et **1001/TCP**. Le port **22/TCP** pour l'accès **ssh** doit également être disponible. Toutes les modifications effectuées doivent persister après un redémarrage.

Si vous y êtes invité **sudo**, utilisez le mot de passe student.

**Important :** l'interface graphique utilisée dans l'environnement de formation en ligne de Red Hat exige également que le port **5900/TCP** reste accessible. Ce port est également connu sous le nom de service **vnc-server**. Si vous avez accidentellement bloqué votre accès au système serverb, vous pouvez soit tenter de vous reconnecter à l'aide de **ssh** à votre machine serverb depuis votre machine workstation, soit réinitialiser la machine serverb. Si vous choisissez de réinitialiser votre machine serverb, vous devez exécuter de nouveau les scripts de configuration pour cet atelier. La configuration de vos machines comprend déjà une zone personnalisée, appelée ROL, qui ouvre ces ports.

1. À partir de workstation, testez l'accès au serveur Web par défaut à l'adresse `http://serverb.lab.example.com` et à l'hôte virtuel à l'adresse `http://serverb.lab.example.com:1001`.
2. Connectez-vous à serverb pour déterminer ce qui empêche l'accès aux serveurs Web.
3. Configurez SELinux pour autoriser le service `httpd` à écouter le port **1001/TCP**.
4. À partir de workstation, testez l'accès au serveur Web par défaut à l'adresse `http://serverb.lab.example.com` et à l'hôte virtuel à l'adresse `http://serverb.lab.example.com:1001`.
5. Connectez-vous à serverb pour déterminer si les ports adéquats sont attribués au pare-feu.

6. Ajoutez le port **1001/TCP** à la configuration permanente de la zone de réseau public. Vérifiez votre configuration.
7. À partir de **workstation**, vérifiez que le serveur Web par défaut à l'adresse **serverb.lab.example.com** renvoie **SERVER B** et que l'hôte virtuel à l'adresse **serverb.lab.example.com:1001** renvoie **VHOST 1**.

## Évaluation

Sur **workstation**, exécutez la commande **lab netsecurity-review grade** pour confirmer que vous avez réussi l'exercice pratique.

```
[student@workstation ~]$ lab netsecurity-review grade
```

## Terminer

Sur **workstation**, exécutez le script **lab netsecurity-review finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netsecurity-review finish
```

L'atelier est maintenant terminé.

## ► SOLUTION

# GESTION DE LA SÉCURITÉ RÉSEAU

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez configurer les paramètres de pare-feu et SELinux pour permettre l'accès à plusieurs serveurs Web exécutés sur serverb.

## RÉSULTATS

Vous devez pouvoir configurer les paramètres de pare-feu et SELinux sur un hôte de serveur Web.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur student à workstation avec le mot de passe student.

À partir de workstation, exécutez la commande **lab netsecurity-review start**. La commande exécute un script de démarrage pour déterminer si l'hôte serverb est accessible sur le réseau.

```
[student@workstation ~]$ lab netsecurity-review start
```

Votre société a décidé d'exécuter une nouvelle application Web. Celle-ci écoute sur les ports **80/TCP** et **1001/TCP**. Le port **22/TCP** pour l'accès **ssh** doit également être disponible. Toutes les modifications effectuées doivent persister après un redémarrage.

Si vous y êtes invité **sudo**, utilisez le mot de passe **student**.

**Important :** l'interface graphique utilisée dans l'environnement de formation en ligne de Red Hat exige également que le port **5900/TCP** reste accessible. Ce port est également connu sous le nom de service **vnc-server**. Si vous avez accidentellement bloqué votre accès au système serverb, vous pouvez soit tenter de vous reconnecter à l'aide de **ssh** à votre machine serverb depuis votre machine workstation, soit réinitialiser la machine serverb. Si vous choisissez de réinitialiser votre machine serverb, vous devez exécuter de nouveau les scripts de configuration pour cet atelier. La configuration de vos machines comprend déjà une zone personnalisée, appelée ROL, qui ouvre ces ports.

- À partir de workstation, testez l'accès au serveur Web par défaut à l'adresse `http://serverb.lab.example.com` et à l'hôte virtuel à l'adresse `http://serverb.lab.example.com:1001`.
  - Testez l'accès au serveur Web `http://serverb.lab.example.com`. Le test échoue. À la fin, le serveur Web doit revenir à **SERVER B**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
curl: (7) Failed to connect to serverb.lab.example.com port 80: Connection refused
```

**CHAPITRE 11 |** Gestion de la sécurité réseau

- 1.2. Testez l'accès à l'hôte virtuel `http://serverb.lab.example.com:1001`. Le test échoue. À la fin, l'hôte virtuel doit revenir à **VHOST 1**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No route to host
```

2. Connectez-vous à `serverb` pour déterminer ce qui empêche l'accès aux serveurs Web.

- 2.1. À partir de `workstation`, ouvrez une session SSH sur `serverb` en tant que `student`. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 2.2. Déterminez si le service `httpd` est actif.

```
[student@serverb ~]$ systemctl is-active httpd
inactive
```

- 2.3. Activez et démarrez le service `httpd`. Le service `httpd` ne parvient pas à démarrer.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
[sudo] password for student: student
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/
lib/systemd/system/httpd.service.
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 2.4. Recherchez les raisons pour lesquelles le service `httpd.service` n'a pas pu démarrer.

```
[student@serverb ~]$ systemctl status httpd.service
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset:
    disabled)
  Active: failed (Result: exit-code) since Thu 2019-04-11 19:25:36 CDT; 19s ago
    Docs: man:httpd.service(8)
    Process: 9615 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,
    status=1/FAILURE)
   Main PID: 9615 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."

Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Starting The Apache HTTP
Server...
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission denied:
AH00072: make_sock: could not bind to address [::]:1001
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission denied:
AH00072: make_sock: could not bind to address 0.0.0.0:1001
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: no listening sockets
available, shutting down
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: AH00015: Unable to open logs
```

**CHAPITRE 11 |** Gestion de la sécurité réseau

```
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

- 2.5. Utilisez la commande **sealert** pour vérifier si SELinux empêche le service `httpd` de se lier au port **1001/TCP**.

```
[student@serverb ~]$ sudo sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----
SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 1001.

***** Plugin bind_ports (99.5 confidence) suggests *****

If you want to allow /usr/sbin/httpd to bind to network port 1001
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 1001
    where PORT_TYPE is one of the following: http_cache_port_t, http_port_t,
    jboss_management_port_t, jboss.messaging_port_t, ntop_port_t, puppet_port_t.

***** Plugin catchall (1.49 confidence) suggests *****

...output omitted...
```

3. Configurez SELinux pour autoriser le service `httpd` à écouter le port **1001/TCP**.

- 3.1. Utilisez la commande **semanage** pour trouver le type de port correct.

```
[student@serverb ~]$ sudo semanage port -l | grep 'http'
http_cache_port_t      tcp  8080, 8118, 8123, 10001-10010
http_cache_port_t      udp  3130
http_port_t            tcp  80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t    tcp  5988
pegasus_https_port_t   tcp  5989
```

- 3.2. Utilisez la commande **semanage** pour lier le port **1001/TCP** au type `http_port_t`.

```
[student@serverb ~]$ sudo semanage port -a -t http_port_t -p tcp 1001
[student@serverb ~]$
```

- 3.3. Confirmez que ce port **1001/TCP** est lié au type de port `http_port_t`.

```
[student@serverb ~]$ sudo semanage port -l | grep '^http_port_t'
http_port_t            tcp  1001, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

- 3.4. Activez et démarrez le service `httpd`.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
```

3.5. Vérifiez l'état d'exécution du service httpd.

```
[student@serverb ~]$ systemctl is-active httpd; systemctl is-enabled httpd
active
enabled
```

3.6. Quittez serverb.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

4. À partir de workstation, testez l'accès au serveur Web par défaut à l'adresse `http://serverb.lab.example.com` et à l'hôte virtuel à l'adresse `http://serverb.lab.example.com:1001`.
  - 4.1. Testez l'accès au serveur Web `http://serverb.lab.example.com`. Le serveur Web doit revenir à **SERVER B**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
SERVER B
```

4.2. Testez l'accès à l'hôte virtuel `http://serverb.lab.example.com:1001`. Le test continue à échouer.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No route to host
```

5. Connectez-vous à serverb pour déterminer si les ports adéquats sont attribués au pare-feu.
  - 5.1. À partir de workstation, connectez-vous à serverb en tant qu'utilisateur student.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

5.2. Vérifiez que la zone de pare-feu par défaut est définie sur **public**.

```
[student@serverb ~]$ firewall-cmd --get-default-zone
public
```

5.3. Si l'étape précédente n'a pas renvoyé `public` en tant que zone par défaut, remédiez-y avec la commande suivante :

```
[student@serverb ~]$ sudo firewall-cmd --set-default-zone public
```

5.4. Déterminez les ports ouverts listés dans la zone de réseau public.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
[sudo] password for student: student
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

6. Ajoutez le port **1001/TCP** à la configuration permanente de la zone de réseau public. Vérifiez votre configuration.

6.1. Ajoutez le port **1001/TCP** à la zone de réseau public.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public \
--add-port=1001/tcp
success
```

6.2. Rechargez la configuration du pare-feu.

```
[student@serverb ~]$ sudo firewall-cmd --reload
success
```

6.3. Vérifiez votre configuration.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 1001/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

6.4. Quittez serverb.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

7. À partir de workstation, vérifiez que le serveur Web par défaut à l'adresse serverb.lab.example.com renvoie **SERVER B** et que l'hôte virtuel à l'adresse serverb.lab.example.com:1001 renvoie **VHOST 1**.

7.1. Testez l'accès au serveur Web `http://serverb.lab.example.com`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com  
SERVER B
```

7.2. Testez l'accès à l'hôte virtuel `http://serverb.lab.example.com:1001`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001  
VHOST 1
```

## Évaluation

Sur workstation, exécutez la commande `lab netsecurity-review grade` pour confirmer que vous avez réussi l'exercice pratique.

```
[student@workstation ~]$ lab netsecurity-review grade
```

## Terminer

Sur workstation, exécutez le script `lab netsecurity-review finish` pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab netsecurity-review finish
```

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Le sous-système **netfilter** permet aux modules du noyau d'examiner chaque paquetage qui traverse le système. Tous les paquets réseau entrants, sortants ou transférés sont examinés.
- L'utilisation de **firewalld** a simplifié la gestion en classifiant tout le trafic réseau en zones. Chaque zone peut avoir sa propre liste de ports et de services. La zone **public** est définie comme zone par défaut.
- Le service **firewalld** est également fourni avec plusieurs services prédéfinis. Ils peuvent être listés en utilisant la commande **firewall-cmd --get-services**.
- La politique SELinux contrôle de manière stricte le trafic réseau. Les ports réseau sont étiquetés. Par exemple, le port **22/TCP** est associé à l'étiquette **ssh\_port\_t**. Lorsqu'un processus veut écouter un port, SELinux vérifie que l'étiquette qui lui est associée autorise la liaison à ce port.
- La commande **semanage** est utilisée pour ajouter, supprimer et modifier des étiquettes.



## CHAPITRE 12

# INSTALLATION DE RED HAT ENTERPRISE LINUX

### PROJET

Installer Red Hat Enterprise Linux sur des serveurs et des machines virtuelles.

### OBJECTIFS

- Installer Red Hat Enterprise Linux sur un serveur.
- Automatiser le processus d'installation à l'aide de Kickstart.
- Installer une machine virtuelle sur votre serveur Red Hat Enterprise Linux à l'aide de Cockpit.

### SECTIONS

- Installation de Red Hat Enterprise Linux (et exercice guidé)
- Automatisation de l'installation avec Kickstart (et exercice guidé)
- Installation et configuration des machines virtuelles (et quiz)

### ATELIER

Installation de Red Hat Enterprise Linux

# INSTALLATION DE RED HAT ENTERPRISE LINUX

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir installer Red Hat Enterprise Linux sur un serveur.

## SÉLECTION DU SUPPORT D'INSTALLATION

Red Hat fournit plusieurs supports d'installation que vous pouvez télécharger sur le site Web du portail client à l'aide de votre abonnement actif.

- Un DVD binaire contenant Anaconda, le programme d'installation Red Hat Enterprise Linux et les dépôts de paquetages BaseOS et AppStream. Ces dépôts contiennent les paquetages nécessaires pour terminer l'installation sans matériel supplémentaire.
- Une image ISO de démarrage contenant Anaconda, mais qui nécessite un réseau configuré pour accéder aux dépôts de paquetages disponibles via HTTP, FTP ou NFS.
- Une image QCOW2 contenant un disque système prédéfini prêt à être déployé en tant que machine virtuelle dans des environnements de cloud ou d'entreprise virtuels. QCOW2 (*QEMU Copy On Write*) est le format d'image standard utilisé par Red Hat.

Red Hat fournit un support d'installation pour quatre architectures de processeur prises en charge : x86 64 bits (AMD et Intel), IBM Power Systems (Little Endian), IBM Z et ARM 64 bits.

Après le téléchargement, gravez le DVD ou l'image ISO de démarrage sur un support physique, copiez-les sur un lecteur flash USB ou similaire, ou publiez-les à partir d'un serveur réseau pour une utilisation automatisée de Kickstart.

## Création d'images avec Composer

Composer est un nouvel outil disponible dans RHEL 8. Pour les cas d'utilisation spécialisés, Composer permet aux administrateurs de créer des images système personnalisées à déployer sur des plateformes cloud ou dans des environnements virtuels.

Composer utilise la console Web graphique Cockpit. Il peut également être invoqué à partir d'une ligne de commande à l'aide de la commande **composer-cli**.

## INSTALLATION MANUELLE AVEC ANACONDA

À l'aide du DVD binaire ou de l'ISO de démarrage, les administrateurs peuvent installer un nouveau système RHEL sur un serveur nu ou une machine virtuelle. Le programme Anaconda prend en charge deux méthodes d'installation :

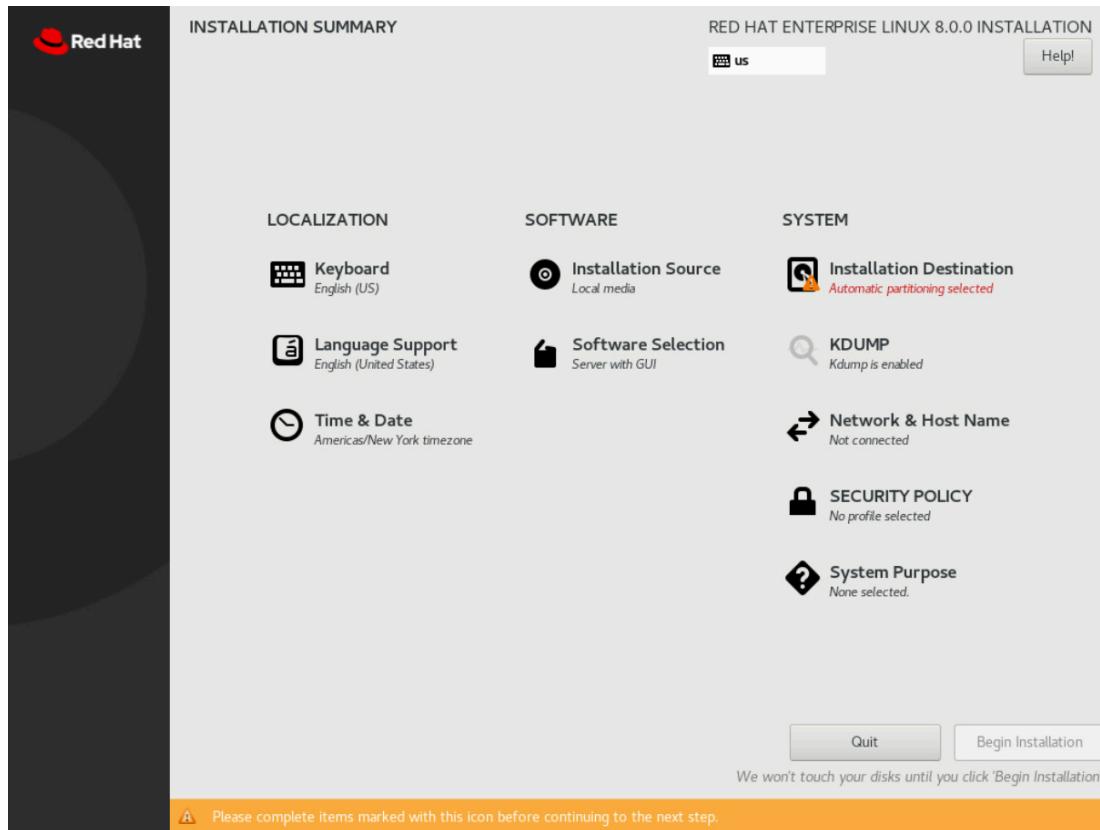
- L'installation manuelle interagit avec l'utilisateur pour demander comment Anaconda doit installer et configurer le système.
- L'installation automatisée utilise un fichier *Kickstart* qui indique à Anaconda comment installer le système. Une section ultérieure décrit plus en détail les installations Kickstart.

## Installation de RHEL avec l'interface graphique

Lorsque vous démarrez le système à partir du DVD binaire ou de l'image ISO de démarrage, Anaconda démarre en tant qu'application graphique.

Sur l'écran Welcome to Red Hat Enterprise Linux 8, sélectionnez la langue à utiliser lors de l'installation. Cela définit également la langue par défaut du système après l'installation. Les utilisateurs individuels peuvent sélectionner la langue qu'ils souhaitent pour leur compte après l'installation.

Anaconda présente la fenêtre Installation Summary, l'emplacement central pour personnaliser les paramètres avant de commencer l'installation.



**Figure 12.1: Fenêtre Installation Summary**

Dans cette fenêtre, configurez les paramètres d'installation en sélectionnant les icônes dans n'importe quel ordre. Sélectionnez l'élément à afficher ou à modifier. Dans n'importe quel élément, cliquez sur Done pour revenir à cet écran central.

Anaconda marque les éléments obligatoires avec un symbole d'avertissement sous forme de triangle et un message. La barre d'état orange en bas de l'écran vous rappelle que ces éléments obligatoires doivent être complétés avant que l'installation puisse commencer.

Complétez les éléments suivants selon vos besoins :

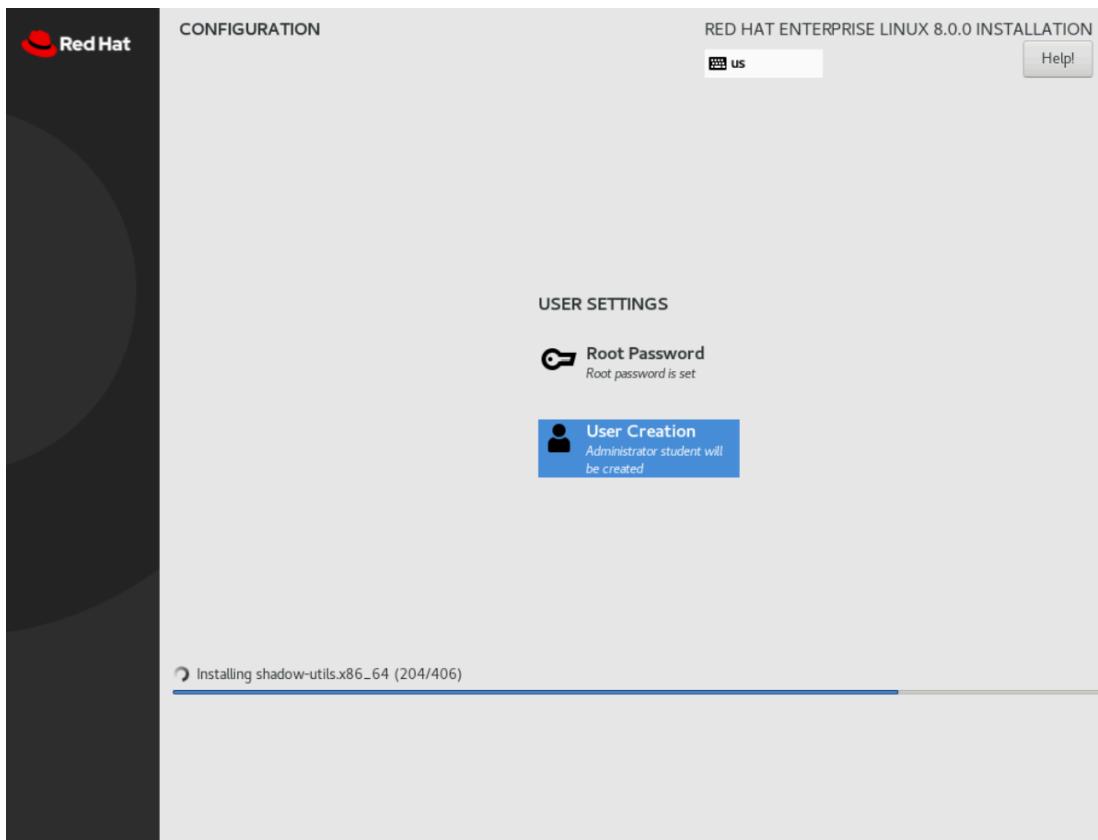
- Keyboard : ajoutez des dispositions de clavier supplémentaires.
- Language Support : sélectionnez d'autres langues à installer.

- Time & Date : sélectionnez la ville de l'emplacement du système en cliquant sur la carte interactive ou en utilisant le menu déroulant. Spécifiez un fuseau horaire, même si vous utilisez le protocole NTP (*Network Time Protocol*).
- Installation Source : indiquez l'emplacement du paquetage source dont Anaconda a besoin pour l'installation. Si vous utilisez le DVD binaire, le champ source de l'installation fait déjà référence au DVD.
- Software Selection : sélectionnez l'environnement de base à installer, ainsi que tous les autres modules complémentaires. L'environnement Minimal Install installe uniquement les paquetages essentiels à l'exécution de Red Hat Enterprise Linux.
- Installation Destination : sélectionnez et partitionnez les disques sur lesquels Red Hat Enterprise Linux sera installé. Cet élément exige de l'administrateur une bonne compréhension des schémas de partitionnement et des critères de sélection des systèmes de fichiers. La case d'option de partitionnement automatique permet d'allouer les périphériques de stockage sélectionnés en utilisant toute la place disponible.
- KDUMP : Kdump est une fonctionnalité du noyau qui collecte le contenu de la mémoire système lorsque le noyau tombe en panne. Les ingénieurs Red Hat peuvent analyser un kdump pour identifier la cause d'un incident. Utilisez cet élément Anaconda pour activer ou désactiver Kdump.
- Network & Host Name : les connexions réseau détectées sont listées dans le volet de gauche. Sélectionnez une connexion pour afficher ses détails. Pour configurer la connexion réseau sélectionnée, cliquez sur Configure.
- SECURITY POLICY : en activant un profil de politique de sécurité, tel que le profil *Payment Card Industry Data Security Standard (PCI DSS)*, Anaconda applique des restrictions et des recommandations, définies par le profil sélectionné, lors de l'installation.
- System Purpose : nouvelle fonction d'installation qui attribue des autorisations système actives en fonction de l'utilisation prévue du système.

Une fois la configuration de l'installation terminée et tous les avertissements résolus, cliquez sur Begin Installation. Un clic sur Quit permet d'abandonner l'installation sans appliquer les modifications éventuelles au système.

Pendant l'installation du système, complétez les éléments suivants lorsqu'ils s'affichent :

- Root Password : le programme d'installation vous invite à définir un mot de passe root. L'étape finale du processus d'installation ne peut pas se poursuivre tant que vous n'avez pas défini un mot de passe root.
- User Creation : créez un compte non root facultatif. Il est recommandé de maintenir un compte local à usage général. Vous pouvez également créer des comptes une fois l'installation terminée.



**Figure 12.2: Définition du mot de passe root et création d'un utilisateur**

Une fois l'installation terminée, cliquez sur Reboot. Anaconda affiche l'écran Initial Setup, si un bureau graphique a été installé. Acceptez les informations de licence et enregistrez éventuellement le système auprès du gestionnaire d'abonnements. Vous pouvez ignorer l'enregistrement du système et l'exécuter plus tard.

## Résolution des problèmes liés à l'installation

Lors d'une installation Red Hat Enterprise Linux 8, Anaconda fournit deux consoles virtuelles. La première de ces consoles comprend cinq fenêtres fournies par le multiplexeur de terminal du logiciel **tmux**. Vous pouvez accéder à cette console avec **Ctrl+Alt+F1**. La deuxième console virtuelle, qui s'affiche par défaut, correspond à l'interface graphique Anaconda. Vous pouvez y accéder avec **Ctrl+Alt+F6**.

Dans la première console virtuelle, **tmux** fournit une invite du shell dans la deuxième fenêtre. Vous pouvez l'utiliser pour saisir des commandes permettant d'examiner le système et d'en résoudre les problèmes pendant que l'installation se poursuit. Les autres fenêtres fournissent des messages de diagnostic, des journaux et d'autres informations.

Le tableau suivant liste les combinaisons de touches permettant d'accéder aux consoles virtuelles et aux fenêtres **tmux**. Dans le cas de **tmux**, les raccourcis clavier s'effectuent en deux actions : appuyez et relâchez **Ctrl+b** et appuyez ensuite sur la touche numérique de la fenêtre à laquelle vous souhaitez accéder. Avec **tmux**, vous pouvez aussi utiliser **Alt+Tab** pour faire passer la sélection actuelle d'une fenêtre à l'autre.

COMBINAISON DE touches	CONTENU
<b>Ctrl+Alt+F1</b>	Accéder au multiplexeur terminal <b>tmux</b> .
<b>Ctrl+b 1</b>	Dans <b>tmux</b> , accéder à la page d'information principale du processus d'installation.
<b>Ctrl+b 2</b>	Dans <b>tmux</b> , fournir un shell racine. Anaconda stocke les fichiers journaux d'installation dans le fichier <b>/tmp</b> .
<b>Ctrl+b 3</b>	Dans <b>tmux</b> , afficher le contenu du fichier <b>/tmp/anaconda.log</b> .
<b>Ctrl+b 4</b>	Dans <b>tmux</b> , afficher le contenu du fichier <b>/tmp/storage.log</b> .
<b>Ctrl+b 5</b>	Dans <b>tmux</b> , afficher le contenu du fichier <b>/tmp/program.log</b> .
<b>Ctrl+Alt+F6</b>	Accéder à l'interface graphique d'Anaconda.



### NOTE

Aux fins de compatibilité avec les versions plus récentes de Red Hat Enterprise Linux, les consoles virtuelles de **Ctrl+Alt+F2** à **Ctrl+Alt+F5** présentent également des shells root lors de l'installation.



### RÉFÉRENCES

Pour plus d'informations, reportez-vous au guide *Installing and deploying RHEL* à l'adresse  
[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/performing\\_a\\_standard\\_rhel\\_installation/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_a_standard_rhel_installation/index)

## ► EXERCICE GUIDÉ

# INSTALLATION DE RED HAT ENTERPRISE LINUX

Dans cet exercice, vous allez réinstaller un de vos serveurs avec une installation minimale de Red Hat Enterprise Linux.

## RÉSULTATS

Vous devez pouvoir installer manuellement Red Hat Enterprise Linux 8.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab installing-install start**. Cette commande exécute un script de démarrage pour déterminer si la machine **servera** est accessible sur le réseau. Elle ajoute également une nouvelle entrée dans le menu GRUB2 pour démarrer **servera** dans le support d'installation.

```
[student@workstation ~]$ lab installing-install start
```

- ▶ 1. Accédez à la console **servera** et redémarrez le système sur le support d'installation.
  - 1.1. Localisez l'icône de la console **servera**, en fonction de votre environnement de formation. Ouvrez la console.
  - 1.2. Pour redémarrer, envoyez une commande **Ctrl+Alt+Suppr** à votre système en utilisant le moyen adéquat : clavier, console virtuelle ou entrée de menu.
  - 1.3. Lorsque le menu du chargeur de démarrage s'affiche, sélectionnez **Install Red Hat Enterprise Linux 8**.
  - 1.4. Patientez jusqu'à l'affichage de la fenêtre de sélection de la langue.
- ▶ 2. Conservez la langue sélectionnée par défaut et cliquez sur **Continuer**.
- ▶ 3. Utilisez le partitionnement automatique sur le disque **/dev/vda**.
  - 3.1. Cliquez sur **Installation Destination**.
  - 3.2. Cliquez sur le premier disque, **vda**, pour le sélectionner. Cliquez sur **Done** pour utiliser l'option par défaut du partitionnement automatique.
  - 3.3. Dans la fenêtre **Installation Options**, cliquez **Reclaim space**. Dans la mesure où le disque **/dev/vda** a déjà des partitions et des systèmes de fichiers de l'installation précédente, cette sélection vous permet de nettoyer le disque pour la nouvelle installation. Dans la fenêtre **Reclaim Disk Space**, cliquez sur **Delete all**, puis sur **Reclaim space**.

- **4.** Définissez le nom d'hôte du serveur sur `serverc.lab.example.com`, et vérifiez la configuration de l'interface réseau.
- 4.1. Cliquez sur Network & Host Name.
  - 4.2. Dans le champ Host Name, entrez **servera.lab.example.com**, puis cliquez sur Apply.
  - 4.3. Cliquez sur Configure, puis sur l'onglet IPv4 Settings.
  - 4.4. Vérifiez que les paramètres du réseau sont corrects. L'adresse IP est `172.25.250.10`, le masque de réseau est 24, et la passerelle et le serveur de noms sont réglés sur `172.25.250.254`. Cliquez sur Save.
  - 4.5. Confirmez que l'interface réseau est activée en définissant ON/OFF sur **ON**.
  - 4.6. Cliquez sur Done.
- **5.** Définissez le champ Installation Source sur `http://content.example.com/rhel8.0/x86_64/dvd`.
- 5.1. Cliquez sur Installation Source.
  - 5.2. Dans le champ http://, tapez **content.example.com/rhel8.0/x86\_64/dvd**
  - 5.3. Cliquez sur Done.
- **6.** Sélectionnez le logiciel nécessaire pour exécuter une installation minimale.
- 6.1. Cliquez sur Software Selection.
  - 6.2. Sélectionnez Minimal Install dans la liste Base Environment.
  - 6.3. Cliquez sur Done.
- **7.** Configurez la fonction du système.
- 7.1. Cliquez sur System Purpose.
  - 7.2. Sélectionnez un rôle de Red Hat Enterprise Linux Server.
  - 7.3. Sélectionnez un niveau de contrat de niveau de service Self-Support.
  - 7.4. Sélectionnez l'utilisation Development/Test.
  - 7.5. Cliquez sur Done.
- **8.** Cliquez sur Begin Installation.
- **9.** Lors de l'installation, définissez le mot de passe `root` sur `redhat`.
- 9.1. Cliquez sur Root Password.
  - 9.2. Saisissez **redhat** dans le champ Root Password.
  - 9.3. Saisissez **redhat** dans le champ Confirm.
  - 9.4. Le mot de passe est faible, vous devez donc cliquer sur Done deux fois.

- 10. Pendant la progression de l'installation, ajoutez l'utilisateur **student**.
- 10.1. Cliquez sur User Creation.
  - 10.2. Saisissez **student** dans le champ Full Name.
  - 10.3. Vérifiez Make this user administrator de sorte que **student** puisse utiliser **sudo** pour exécuter des commandes en tant que **root**.
  - 10.4. Saisissez **student** dans le champ Password.
  - 10.5. Saisissez **student** dans le champ Confirm Password.
  - 10.6. Le mot de passe est faible, vous devez donc cliquer sur Done deux fois.
- 11. Une fois l'installation terminée, cliquez sur Reboot.
- 12. Lorsque le système affiche l'invite de connexion, connectez-vous en tant que **student** avec le mot de passe **student**.

## Fin

Utilisez la méthode appropriée à votre environnement de classe pour réinitialiser votre machine serveur.

L'exercice guidé est maintenant terminé.

# AUTOMATISATION DE L'INSTALLATION AVEC KICKSTART

---

## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir réaliser les tâches suivantes :

- Décrire les concepts et l'architecture sur lesquels repose Kickstart.
- Créer un fichier Kickstart avec le site Web Kickstart Generator.
- Modifier un fichier Kickstart existant avec un éditeur de texte et vérifier sa syntaxe avec **ksvalidator**.
- Publier un fichier Kickstart dans le programme d'installation.
- Effectuer une installation Kickstart réseau.

## CRÉATION D'UN PROFIL KICKSTART

Vous pouvez automatiser l'installation de Red Hat Enterprise Linux à l'aide d'une fonction appelée **Kickstart**. En utilisant Kickstart, vous spécifiez tout ce dont Anaconda a besoin pour effectuer une installation, notamment le partitionnement de disque, la configuration de l'interface réseau, la sélection du paquetage et d'autres paramètres, dans un fichier texte Kickstart. En référençant le fichier texte, Anaconda effectue l'installation sans autre intervention de l'utilisateur.



### NOTE

La fonction Kickstart de Red Hat Enterprise Linux est similaire à la fonction Jumpstart d'Oracle Solaris ou à l'utilisation d'un fichier de réponses d'installation sans assistance de Microsoft Windows.

Les fichiers Kickstart commencent par une liste de commandes qui définissent comment installer la machine cible. Les lignes précédées d'un caractère **#** sont des commentaires qui doivent être ignorés par le programme d'installation. Les sections supplémentaires commencent par une **directive**, qui se distingue par un caractère initial **%** et qui se conclut en fin de ligne par la directive **%end**.

La section **%packages** indique le logiciel à installer sur le système cible. Spécifiez les paquetages individuels par nom (sans numéro de version). Les groupes de paquetages, spécifiés par leur nom ou leur ID, commencent par un caractère **@**. Les groupes d'environnement (groupes de groupes de paquetages) commencent par les caractères **@^**. Spécifiez des modules, des flux et des profils avec la syntaxe **@module:stream/profile**.

Les groupes comportent des composants obligatoires, par défaut et facultatifs. Normalement, Kickstart installent les composants obligatoires et par défaut. Pour exclure un paquetage ou un groupe de paquetages de l'installation, faites-le précéder d'un caractère **-**. Toutefois, les paquetages ou groupes de paquetages exclus peuvent toujours être installés s'ils constituent des dépendances obligatoires d'autres paquetages demandés.

Une configuration Kickstart utilise couramment deux sections supplémentaires, **%pre** et **%post**, qui contiennent des commandes de script shell permettant de configurer davantage le système.

Le script **%pre** est exécuté avant tout partitionnement de disque. En règle générale, cette section est utilisée uniquement si des actions sont nécessaires pour reconnaître ou initialiser un périphérique avant le partitionnement du disque. Le script **%post** est exécuté une fois l'installation terminée.

Vous devez spécifier les commandes Kickstart principales avant les sections **%pre**, **%post** et **%packages**, mais sinon, vous pouvez placer ces sections dans n'importe quel ordre dans le fichier.

## COMMANDES DU FICHIER KICKSTART

### Commandes d'installation

Définissez la source d'installation et le mode d'installation. Chaque commande est suivie d'un exemple.

- **url** : spécifie l'adresse URL cible du support d'installation.

```
url --url="http://classroom.example.com/content/rhel8.0/x86_64/dvd/"
```

- **repo** : spécifie où trouver des paquetages supplémentaires pour l'installation. Cette option doit pointer vers un dépôt **yum** valide.

```
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

- **text** : force l'installation en mode texte.
- **vnc** : autorise la visualisation à distance de l'installation graphique sur VNC.

```
vnc --password=redhat
```

### Commandes de partitionnement

Définissez les périphériques et le schéma de partitionnement à utiliser.

- **clearpart** : supprime les partitions du système avant la création d'autres partitions. Par défaut, aucune partition n'est supprimée.

```
clearpart --all --drives=sda,sdb --initlabel
```

- **part** : spécifie la taille, le format et le nom d'une partition.

```
part /home --fstype=ext4 --label=homes --size=4096 --maxsize=8192 --grow
```

- **autopart** : crée automatiquement une partition racine, une partition swap et une partition de démarrage appropriée pour l'architecture. Sur des disques suffisamment grands, une partition `/home` est également créée.
- **ignoredisk** : contrôle l'accès d'Anaconda aux disques connectés au système.

```
ignoredisk --drives=sdc
```

- **bootloader** : définit l'emplacement d'installation du chargeur de démarrage.

```
bootloader --location=mbr --boot-drive=sda
```

- **volgroup, logvol** : crée des groupes de volumes et des volumes logiques LVM.

```
part pv.01 --size=8192
volgroup myvg pv.01
logvol / --vgname=myvg --fstype=xfs --size=2048 --name=rootvol --grow
logvol /var --vgname=myvg --fstype=xfs --size=4096 --name=varvol
```

- **zerombr** : initialise les disques dont le formatage n'est pas reconnu.

## Commandes de réseau

Définissez les fonctionnalités réseau utilisées par l'hôte.

- **network** : configure les informations réseau du système cible. Active les périphériques réseau dans l'environnement d'installation.

```
network --device=eth0 --bootproto=dhcp
```

- **firewall** : définit la configuration du pare-feu sur le système cible.

```
firewall --enabled --service=ssh,http
```

## Commandes liées à l'emplacement et à la sécurité

Configurez les paramètres liés à la sécurité, à la langue et aux régions.

- **lang** : définit la langue à utiliser lors de l'installation et la langue par défaut du système installé. Obligatoire.

```
lang en_US.UTF-8
```

- **keyboard** : définit le type de clavier du système. Obligatoire.

```
keyboard --vckeymap=us --xlayouts=''
```

- **timezone** : définit le fuseau horaire, les serveurs NTP et l'éventuelle utilisation du temps UTC par l'horloge matérielle.

```
timezone --utc --ntp servers=time.example.com Europe/Amsterdam
```

- **authselect** : configure les options d'authentification. Les options reconnues par **authselect** sont valides pour cette commande. Voir authselect(8).

- **rootpw** : définit le mot de passe **root** initial.

```
rootpw --plaintext redhat
or
rootpw --iscrypted $6$KUnFfrTz08jv.PiH$Y1Bb0tXBkWzoMuRfb0.SpbQ....XDR1UuchoMG1
```

## CHAPITRE 12 | Installation de Red Hat Enterprise Linux

- **selinux** : définit le mode SELinux pour le système installé.

```
selinux --enforcing
```

- **services** : modifie l'ensemble de services par défaut qui s'exécute sous la cible **systemd** par défaut.

```
services --disabled=network,iptables,ip6tables --enabled=NetworkManager,firewalld
```

- **group, user** : crée un groupe ou un utilisateur locaux sur le système.

```
group --name=admins --gid=10001
user --name=jdoe --gecos="John Doe" --groups=admins --password=changeme --
plaintext
```

## Commandes diverses

Configurez divers éléments liés à la journalisation lors de l'installation et à l'état d'alimentation de l'hôte à la fin.

- **logging** : définit le mode de journalisation d'Anaconda lors de l'installation.

```
logging --host=loghost.example.com --level=info
```

- **firstboot** : s'il est activé, l'agent de configuration est lancé au premier démarrage du système. Le paquetage *initial-setup* doit être installé.

```
firstboot --disabled
```

- **reboot, poweroff, halt** : spécifiez la dernière action à effectuer à la fin de l'installation.



### NOTE

L'utilitaire **ksverdiff** du paquetage *pykickstart* permet d'identifier les différences de syntaxe des fichiers Kickstart entre deux versions de Red Hat Enterprise Linux ou Fedora.

Par exemple, **ksverdiff -f RHEL7 -t RHEL8** identifie les changements de syntaxe entre RHEL 7 et RHEL 8. Les versions disponibles sont listées en haut du fichier */usr/lib/python3.6/site-packages/pykickstart/version.py*.

## EXEMPLE DE FICHIER KICKSTART

La première partie du fichier se compose des commandes d'installation, telles que le partitionnement du disque et la source d'installation.

```
#version=RHEL8
ignoredisk --only-use=vda
# System bootloader configuration
bootloader --append="console=ttyS0 console=ttyS0,115200n8 no_timer_check
net.ifnames=0 crashkernel=auto" --location=mbr --timeout=1 --boot-drive=vda
# Clear the Master Boot Record
```

```

zerombr
# Partition clearing information
clearpart --all --initlabel
# Use text mode install
text
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/
x86_64/dvd/AppStream/
# Use network installation
url --url="http://classroom.example.com/content/rhel8.0/x86_64/dvd/"
# Keyboard layouts
# old format: keyboard us
# new format:
keyboard --vckeymap=us --xlayouts=''
# System language
lang en_US.UTF-8
# Root password
rootpw --plaintext redhat
# System authorization information
auth --enablesystem --passalgo=sha512
# SELinux configuration
selinux --enforcing
firstboot --disable
# Do not configure the X Window System
skipx
# System services
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chrony"
# System timezone
timezone America/New_York --isUtc
# Disk partitioning information
part / --fstype="xfs" --ondisk=vda --size=10000

```

La deuxième partie contient la section **%packages** qui indique quels paquetages et groupes de paquetages doivent être installés, ou non.

```

%packages
@core
chrony
cloud-init
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
-plymouth

%end

```

La dernière partie contient les scripts d'installation **%pre** et **%post**.

```
%post --erroronfail

# For cloud images, 'eth0' _is_ the predictable device name, since
# we don't want to be tied to specific virtual (!) hardware
rm -f /etc/udev/rules.d/70*
ln -s /dev/null /etc/udev/rules.d/80-net-name-slot.rules

# simple eth0 config, again not hard-coded to the build hardware
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
EOF

%end
```



### NOTE

L'absence de valeurs obligatoires dans un fichier Kickstart fait passer le programme d'installation en mode interactif, le temps d'obtenir une réponse de la part de l'administrateur, ou provoque l'abandon de l'ensemble de la procédure d'installation.

## PROCÉDURE D'INSTALLATION KICKSTART

Pour automatiser correctement l'installation de Red Hat Enterprise Linux, effectuez les étapes suivantes :

1. Créez un fichier Kickstart.
2. Publiez un fichier Kickstart dans le programme d'installation.
3. Démarrez Anaconda et faites-le pointer vers le fichier de configuration Kickstart.

## CRÉATION D'UN FICHIER KICKSTART

Utilisez l'une de ces méthodes pour créer un fichier Kickstart :

- Utilisez le site Web Kickstart Generator.
- Utilisez un éditeur de texte.

Le site Web de Kickstart Generator à l'adresse <https://access.redhat.com/labs/kickstartconfig/> comporte des boîtes de dialogue pour les entrées utilisateur et crée un fichier texte de directives Kickstart avec les choix de l'utilisateur. Chaque boîte de dialogue correspond aux éléments configurables dans le programme d'installation d'Anaconda.

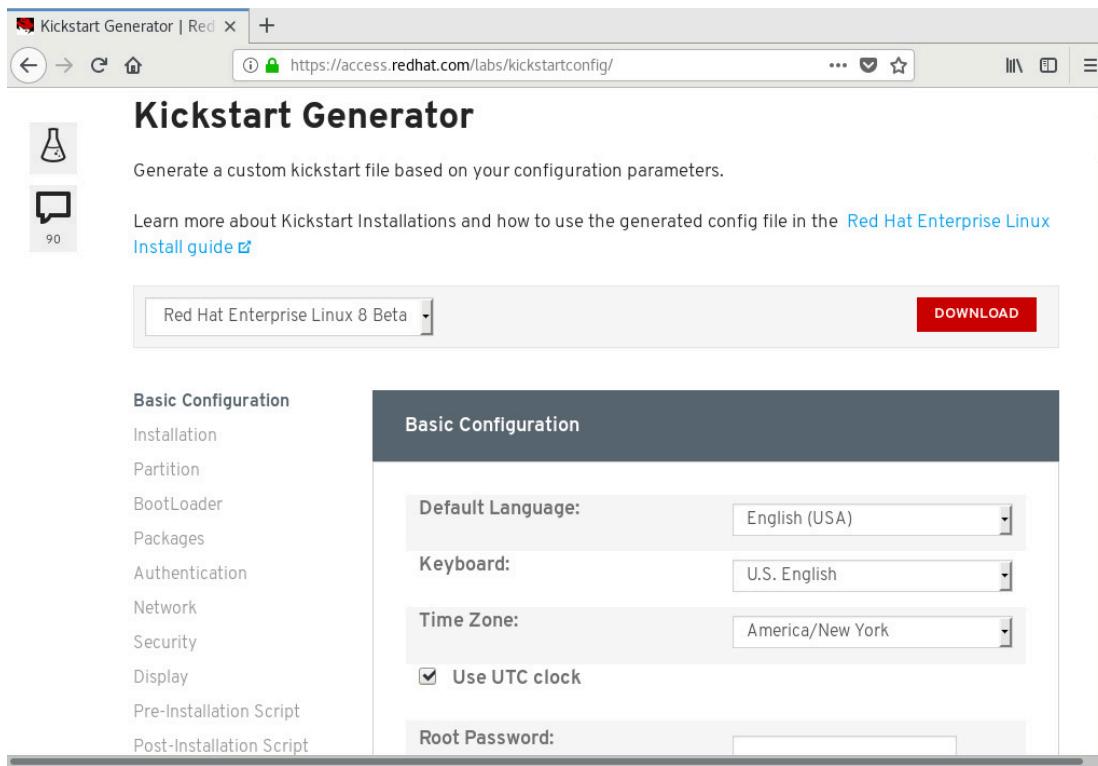


Figure 12.3: Configuration de base avec Kickstart Generator

**NOTE**

Au moment de la rédaction de ce document, le site Web de Kickstart Generator ne proposait pas Red Hat Enterprise Linux 8 comme option de menu. Red Hat Enterprise Linux 8 Beta était une sélection valide.

La création d'un fichier Kickstart à partir de zéro est généralement trop complexe, mais la modification d'un fichier Kickstart existant est courante et utile. Chaque installation crée un fichier **/root/anaconda-ks.cfg** contenant les directives Kickstart utilisées lors de l'installation. Ce fichier constitue un bon point de départ pour créer manuellement un fichier Kickstart.

**ksvalidator** est un utilitaire qui vérifie les erreurs de syntaxe dans un fichier Kickstart. Il vérifie que les mots-clés et options sont utilisés correctement, mais il ne valide ni les URL, ni les paquetages individuels, ni les groupes de paquetages, ni aucune partie des scripts **%post** ou **%pre**. Par exemple, si l'instruction **firewall --disabled** est mal orthographiée, **ksvalidator** peut générer l'une des erreurs suivantes :

```
[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
The following problem occurred on line 10 of the kickstart file:

Unknown command: firewall

[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
The following problem occurred on line 10 of the kickstart file:

no such option: --dsabled
```

Le paquetage **pykickstart** fournit **ksvalidator**.

## PUBLICATION DU FICHIER KICKSTART À L'USAGE D'ANACONDA

Mettez le fichier Kickstart à la disposition du programme d'installation en le plaçant à l'un des emplacements suivants :

- sur un serveur réseau disponible au moment de l'installation via FTP, HTTP ou NFS ;
- sur un disque USB disponible ou CD-ROM ;
- sur un disque dur local sur le système à installer.

Le programme d'installation doit accéder au fichier Kickstart pour démarrer une installation automatisée. La méthode d'automatisation la plus courante utilise un serveur réseau tel qu'un serveur FTP, Web ou NFS. Les serveurs réseau facilitent la maintenance des fichiers Kickstart, car les modifications peuvent être apportées une seule fois, puis servir immédiatement à des installations futures.

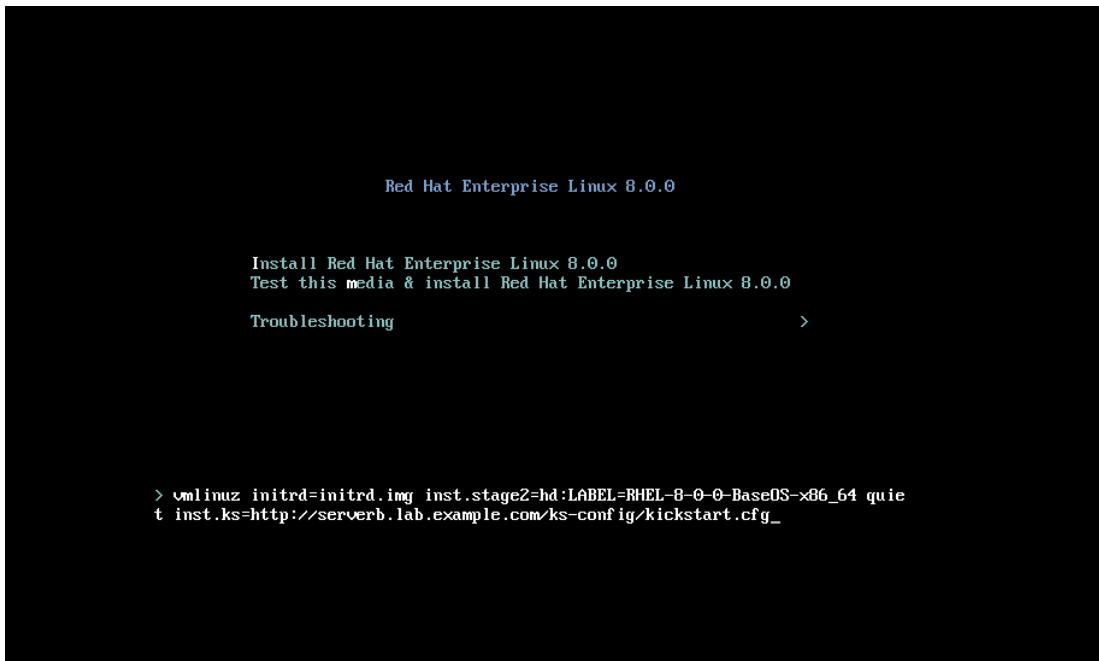
La fourniture de fichiers Kickstart sur USB ou CD-ROM est également pratique. Le fichier Kickstart peut être incorporé sur le support de démarrage utilisé pour le démarrage de l'installation. Toutefois, lorsque le fichier Kickstart est modifié, vous devez générer un nouveau support d'installation.

La fourniture du fichier Kickstart sur un disque local vous permet de reconstruire rapidement un système.

## DÉMARRAGE D'ANACONDA ET POINTAGE DE CE DERNIER VERS LE FICHIER KICKSTART

Une fois qu'une méthode Kickstart a été choisie, le programme d'installation doit être informé de l'emplacement du fichier Kickstart en transmettant le paramètre **inst.ks=LOCATION** au noyau d'installation. Voici quelques exemples :

- `inst.ks=http://serveur/rep/fichier`
- `inst.ks=ftp://serveur/rep/fichier`
- `inst.ks=nfs:serveur:/rep/fichier`
- `inst.ks=hd:peripherique:/rep/fichier`
- `inst.ks=cdrom:peripherique`



**Figure 12.4: Spécification de l'emplacement du fichier Kickstart lors de l'installation**

Pour les installations de machines virtuelles à l'aide du Virtual Machine Manager ou de **virt-manager**, l'URL Kickstart peut être spécifié dans un champ situé sous URL Options. Lors de l'installation de machines physiques, démarrez à partir du support d'installation et appuyez sur la touche **Tab** pour interrompre le processus de démarrage. Ajoutez un paramètre **inst.ks=LOCATION** au noyau d'installation.



## RÉFÉRENCES

Chapitre *Kickstart installation basics* de *Performing an advanced RHEL installation* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/performing\\_an\\_advanced\\_rhel\\_installation/kickstart-installation-basics\\_installing-rhel-as-an-experienced-user#kickstart-installation-basics\\_installing-rhel-as-an-experienced-user](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-installation-basics_installing-rhel-as-an-experienced-user)

Section *Kickstart commands for installation program configuration and flow control* dans *Appendix B. Kickstart commands and options reference* de *Performing an advanced RHEL installation* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/performing\\_an\\_advanced\\_rhel\\_installation/kickstart-installation-basics\\_installing-rhel-as-an-experienced-user#kickstart-commands-for-installation-program-configuration-and-flow-control\\_kickstart-commands-and-options-reference](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-commands-for-installation-program-configuration-and-flow-control_kickstart-commands-and-options-reference)

Chapitre *Boot options* de *Performing an advanced RHEL installation* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/performing\\_an\\_advanced\\_rhel\\_installation/kickstart-installation-basics\\_installing-rhel-as-an-experienced-user#kickstart-and-advanced-boot-options\\_installing-rhel-as-an-experienced-user](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-and-advanced-boot-options_installing-rhel-as-an-experienced-user)

## ► EXERCICE GUIDÉ

# AUTOMATISATION DE L'INSTALLATION AVEC KICKSTART

Dans cet atelier, vous allez créer un fichier Kickstart et en valider la syntaxe.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un fichier kickstart.
- Utiliser **ksvalidator** pour valider la syntaxe du fichier kickstart.

## AVANT DE COMMENCER

Connectez-vous en tant qu'utilisateur **student** à **workstation** avec le mot de passe **student**.

Sur **workstation**, exécutez la commande **lab installing-kickstart start**. Cette commande exécute un script de démarrage pour déterminer si la machine **servera** est accessible sur le réseau. Il vérifie également qu'Apache est installé et configuré sur **servera**.

```
[student@workstation ~]$ lab installing-kickstart start
```

- 1. Utilisez la commande **ssh** pour vous connecter à **servera** en tant qu'utilisateur **student**. Les systèmes sont configurés pour utiliser des clés SSH pour l'authentification. Par conséquent, aucun mot de passe n'est requis.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Copiez **/root/anaconda-ks.cfg** sur **servera** dans un fichier appelé **/home/student/kickstart.cfg** modifiable par **student**. Utilisez la commande **sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg** pour copier le contenu de **/root/anaconda-ks.cfg** dans **/home/student/kickstart.cfg**. Si **sudo** demande le mot de passe de l'utilisateur **student**, utilisez **student** comme mot de passe.

```
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

- 3. Apportez les modifications ci-après au fichier **/home/student/kickstart.cfg**.

- 3.1. Commentez la directive **reboot** :

```
#reboot
```

- 3.2. Commentez la commande **repo** pour le dépôt BaseOS. Modifiez la commande **repo** de sorte qu'AppStream pointe vers le dépôt AppStream de la classe :

```
#repo --name="koji-override-0" --baseurl=http://download-node-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.0.0-20190213.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

- 3.3. Modifiez la commande **url** afin de spécifier le support source d'installation HTTP de la classe :

```
url --url="http://classroom.example.com/content/rhel8.0/x86_64/dvd/"
```

- 3.4. Commentez la commande **network** :

```
#network --bootproto=dhcp --device=link --activate
```

- 3.5. Définissez le mot de passe root sur **redhat**. Remplacez la ligne commençant par **rootpw** par la ligne suivante :

```
rootpw --plaintext redhat
```

- 3.6. Supprimez la ligne qui utilise la commande **auth** et ajoutez la ligne **authselect select sssd** pour définir le service **sssd** en tant que source d'identité et d'authentification.

```
authselect select sssd
```

Dans Red Hat Enterprise Linux 8, la commande **authselect** remplace la commande **authconfig**.

- 3.7. Simplifiez la commande **services** afin qu'elle ressemble exactement à ce qui suit :

```
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chronyd"
```

- 3.8. Commentez les commandes **part** et **reqpart**. Ajoutez la commande **autopart** :

```
#reqpart
# Disk partitioning information
#part / --fstype="xfs" --ondisk=vda --size=8000
autopart
```

- 3.9. Supprimez tout le contenu de la section **%post** et de son **%end**. Ajoutez la ligne suivante : **echo "Kickstarted on \$(date)" >> /etc/issue**  
La section **%post** entière doit ressembler à ceci.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

3.10. Simplifiez la spécification des paquetages exactement comme suit :

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
httpd
-plymouth
%end
```

Enregistrez et quittez le fichier quand vous avez terminé.

- ▶ 4. Utilisez la commande **ksvalidator** pour rechercher les erreurs de syntaxe dans le fichier Kickstart.

```
[student@servera ~]$ ksvalidator kickstart.cfg
```

- ▶ 5. Copiez **kickstart.cfg** dans le répertoire **/var/www/html/ks-config**.

```
[student@servera ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

- ▶ 6. Quittez servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

## Fin

Sur workstation, exécutez le script **lab installing-kickstart finish** pour mettre fin à l'exercice.

```
[student@workstation ~]$ lab installing-kickstart finish
```

L'exercice guidé est maintenant terminé.

# INSTALLATION ET CONFIGURATION DES MACHINES VIRTUELLES

---

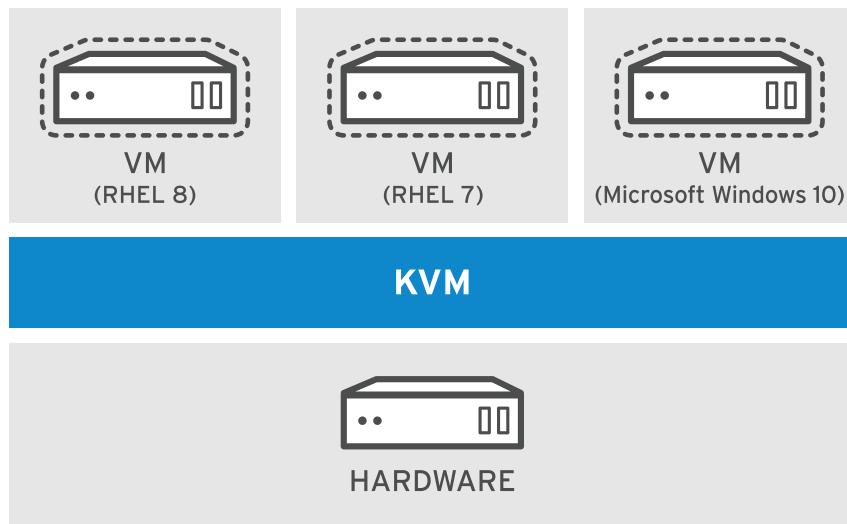
## OBJECTIFS

Après avoir terminé cette section, vous devez pouvoir installer une machine virtuelle sur votre serveur Red Hat Enterprise Linux à l'aide de Cockpit.

## INTRODUCTION À LA VIRTUALISATION KVM

La virtualisation est une fonction qui permet de diviser une machine physique unique en plusieurs *machines virtuelles* (VM), chacune étant capable d'exécuter un système d'exploitation indépendant.

Red Hat Enterprise Linux 8 prend en charge *KVM* (*Kernel-based Virtual Machine*), une solution de virtualisation complète intégrée au noyau Linux standard. KVM peut exécuter plusieurs systèmes d'exploitation Windows et Linux hébergés.



**Figure 12.5: Virtualisation KVM**

Dans Red Hat Enterprise Linux, gérez KVM avec la commande **virsh** ou avec l'outil Virtual Machines de Cockpit.

La technologie de machine virtuelle KVM est disponible dans tous les produits Red Hat, des instances physiques autonomes de Red Hat Enterprise Linux jusqu'à Red Hat OpenStack Platform :

- Les systèmes de matériel physique exécutent Red Hat Enterprise Linux afin de fournir la virtualisation KVM. Red Hat Enterprise Linux est généralement configuré en tant qu'hôte *lourd*, un système qui prend en charge les VM tout en fournissant d'autres services, applications et fonctions de gestion locaux et en réseau.
- *Red Hat Virtualization (RHV)* fournit une interface Web centralisée permettant aux administrateurs de gérer une infrastructure virtuelle complète. Il inclut des fonctionnalités avancées telles que la migration KVM, la redondance et la haute disponibilité. *Red Hat*

*Virtualization Hypervisor* est une version personnalisée de Red Hat Enterprise Linux, dédiée à l'allocation de ressources des machines virtualisées et à leur prise en charge.

- *Red Hat OpenStack Platform (RHOSP)* sert de base pour créer, déployer et mettre à l'échelle un cloud public ou privé.

Red Hat prend en charge les machines virtuelles exécutant ces systèmes d'exploitation :

- Red Hat Enterprise Linux 6 et version ultérieure
- Microsoft Windows 10 et version ultérieure
- Microsoft Windows Server 2016 et version ultérieure

## CONFIGURATION D'UN SYSTÈME PHYSIQUE RED HAT ENTERPRISE LINUX EN TANT QU'HÔTE DE VIRTUALISATION

Les administrateurs peuvent configurer un système Red Hat Enterprise Linux en tant qu'hôte de virtualisation approprié au développement, aux tests, à la formation ou aux besoins d'utilisation simultanée de plusieurs systèmes d'exploitation.

### Installation des outils de virtualisation

Installez le module YUM *virt* pour préparer un système à devenir un hôte de virtualisation.

```
[root@host ~]# yum module list virt
Name           Stream      Profiles          Summary
virt           rhel [d][e]  common [d]       Virtualization module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@host ~]# yum module install virt
...output omitted...
```

### Vérification de la configuration système requise

KVM nécessite soit un processeur Intel avec les extensions Intel VT-x et Intel 64 pour les systèmes x86, ou un processeur AMD avec les extensions AMD-V et AMD64. Pour vérifier votre matériel et la configuration système requise, utilisez la commande **virt-host-validate**.

```
[root@host ~]# virt-host-validate
QEMU: Checking for hardware virtualization : PASS
QEMU: Checking if device /dev/kvm exists : PASS
QEMU: Checking if device /dev/kvm is accessible : PASS
QEMU: Checking if device /dev/vhost-net exists : PASS
QEMU: Checking if device /dev/net/tun exists : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'memory' controller mount-point : PASS
QEMU: Checking for cgroup 'cpu' controller support : PASS
QEMU: Checking for cgroup 'cpu' controller mount-point : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuacct' controller mount-point : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller mount-point : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup 'devices' controller mount-point : PASS
```

```
QEMU: Checking for cgroup 'blkio' controller support      : PASS
QEMU: Checking for cgroup 'blkio' controller mount-point : PASS
QEMU: Checking for device assignment IOMMU support       : PASS
```

Le système doit réussir tous les éléments de validation pour pouvoir être un hôte KVM.

## GESTION DES MACHINES VIRTUELLES AVEC COCKPIT

Le module YUM `virt` fournit la commande `virsh` pour gérer vos machines virtuelles. L'outil Cockpit fournit une interface de console Web pour la gestion KVM et la création de machines virtuelles.

The screenshot shows the Cockpit web interface for managing virtual machines. The left sidebar has a dark theme with the following menu items:

- Virtual Machines - servera.lab.example.com
- System
- Logs
- Networking
- Virtual Machines** (selected)
- Accounts
- Services
- Applications
- Diagnostic Reports
- Kernel Dump
- SELinux
- Software Updates
- Subscriptions

The main content area is titled "Virtual Machines". It shows summary statistics: 2 Storage Pools and 1 Networks. Below this is a table of virtual machines:

Name	Connection	State
dev-rhel8	System	running
prod-rhel7	System	shut off
windows10	System	shut off

For the selected VM "dev-rhel8", detailed configuration options are shown in a modal dialog:

- Overview** tab is active.
- Memory:** 1 GiB
- vCPUs:** 1
- CPU Type:** custom (SandyBridge)
- Emulated Machine:** pc-q35-rhel7.6.0
- Boot Order:** cdrom,disk
- Autostart:** disabled

Actions for the VM include: **Restart**, **Shut Down**, and a red **Delete** button.

Figure 12.6: Gestion des machines virtuelles dans Cockpit

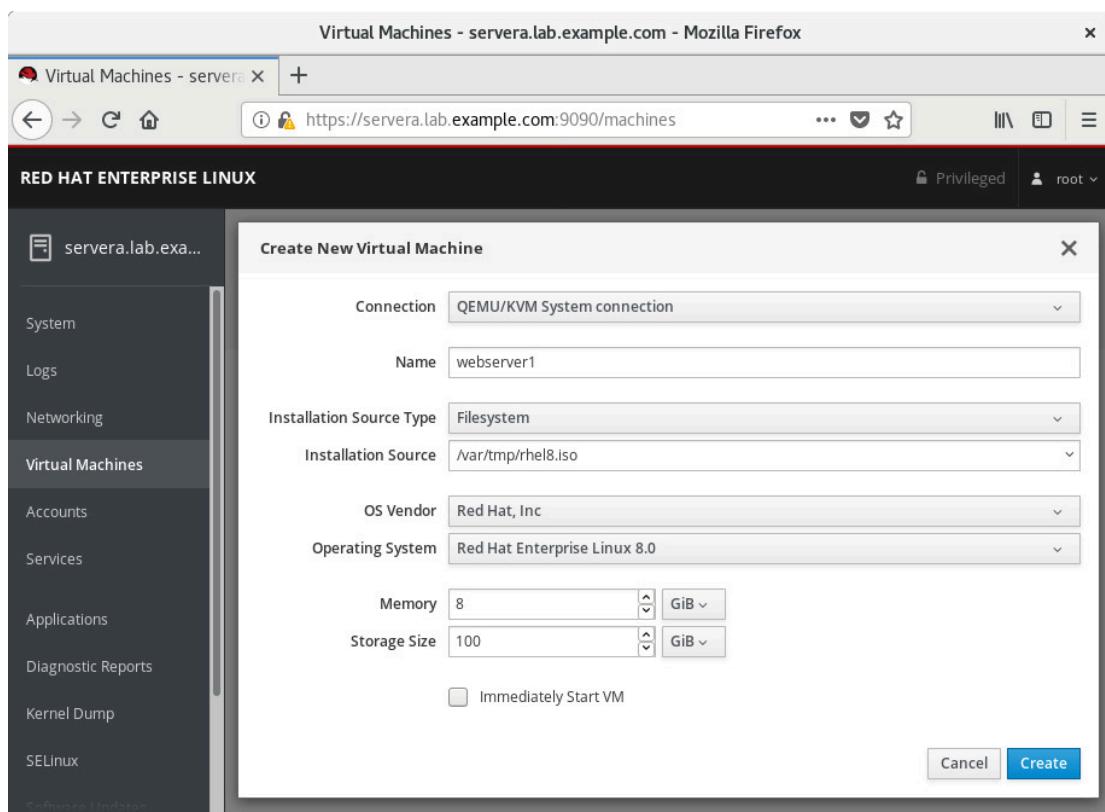
Installez le paquetage `cockpit-machines` afin d'ajouter le menu Virtual Machines à Cockpit.

```
[root@host ~]# yum install cockpit-machines
```

Si Cockpit ne fonctionne pas déjà, démarrez-le et activez-le.

```
[root@host ~]# systemctl enable --now cockpit.socket
```

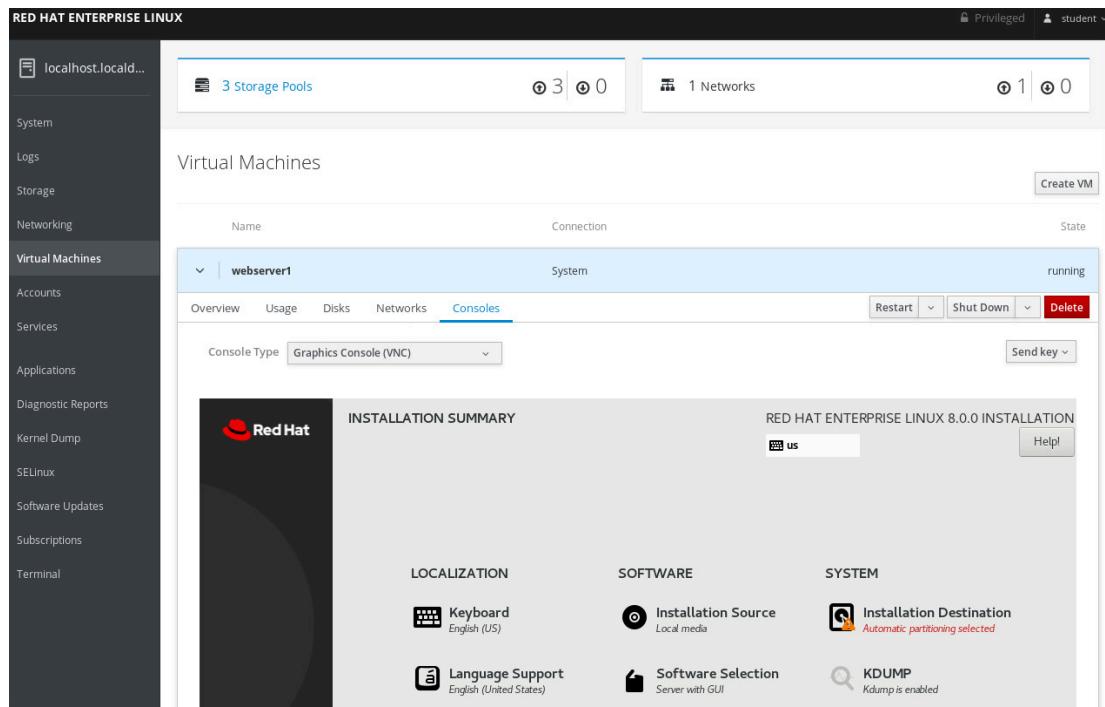
Pour créer une machine virtuelle avec Cockpit, accédez au menu Virtual Machines dans l'interface Web de Cockpit. À partir de là, cliquez sur Create VM et entrez la configuration de la VM dans la fenêtre Create New Virtual Machine.



**Figure 12.7: Crédit d'une machine virtuelle dans Cockpit**

- Name définit un nom de *domaine* pour la configuration de la machine virtuelle. Ce nom n'a aucun lien avec le nom d'hôte réseau que vous donnez au système dans la VM installée.
- Installation Source Type est la méthode permettant de récupérer le fichier ISO d'installation. Les choix incluent le système de fichiers local ou une URL HTTPS, FTP ou NFS.
- Installation Source fournit le chemin d'accès à la source d'installation.
- OS Vendor et Operating System indique le système d'exploitation de la machine virtuelle. La couche de virtualisation présente une émulation matérielle compatible avec le système d'exploitation choisi.
- Memory est la quantité de RAM de la nouvelle machine virtuelle.
- Storage Size est la taille du disque de la nouvelle machine virtuelle. Associez des disques supplémentaires à la machine virtuelle après l'installation.
- Immediately Start VM indique si la VM doit immédiatement démarrer après que vous avez cliqué sur Create.

Cliquez sur Create pour créer la VM et Install pour démarrer l'installation du système d'exploitation. Cockpit affiche la console de machine virtuelle à partir de laquelle vous pouvez installer le système.



**Figure 12.8: Installation du système d'exploitation des machines virtuelles**



## RÉFÉRENCES

Pour plus d'informations, consultez le guide *Configuring and managing virtualization* à l'adresse

[https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html-single/configuring\\_and\\_managing\\_virtualization/index](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_virtualization/index)

## Qu'est-ce que la virtualisation ?

<https://www.redhat.com/en/topics/virtualization/what-is-virtualization>

## ► QUIZ

# INSTALLATION ET CONFIGURATION DES MACHINES VIRTUELLES

Répondez aux questions suivantes en sélectionnant un ou plusieurs éléments :

- 1. Quels sont les systèmes d'exploitation hébergés pris en charge par Red Hat en tant que machines virtuelles KVM? (Choisissez-en trois.)
- a. Fedora 28 et version ultérieure
  - b. Red Hat Enterprise Linux 6 et version ultérieure
  - c. CoreOS Container Linux 2023 et version ultérieure
  - d. Microsoft Windows 7 SP1
  - e. Microsoft Windows 10 et version ultérieure
  - f. Microsoft Windows Server 2016 et version ultérieure
- 2. Quels sont les composants requis pour configurer votre système en tant qu'hôte de virtualisation et gérer les machines virtuelles avec la console Web ? (Choisissez-en deux.)
- a. Le module YUM *virt*
  - b. Le groupe de paquetages Openstack
  - c. Le paquetage *cockpit-machines*
  - d. Le groupe de paquetages *Virtualization Platform*
  - e. Le module YUM *kvm*
  - f. Le paquetage *cockpit-virtualization*
- 3. Quelle commande vérifie que votre système prend en charge la virtualisation ?
- a. grep kvm /proc/cpuinfo
  - b. valsh valider
  - c. virt-host-validate
  - d. rhv-validate
  - e. cockpit-validate
- 4. Quels outils pouvez-vous utiliser pour démarrer et arrêter vos machines virtuelles sur un système Red Hat Enterprise Linux ? (Choisissez-en deux.)
- a. vmctl
  - b. libvirtd
  - c. virsh
  - d. openstack
  - e. Console Web

## ► SOLUTION

# INSTALLATION ET CONFIGURATION DES MACHINES VIRTUELLES

Répondez aux questions suivantes en sélectionnant un ou plusieurs éléments :

- ▶ 1. **Quels sont les systèmes d'exploitation hébergés pris en charge par Red Hat en tant que machines virtuelles KVM? (Choisissez-en trois.)**
  - a. Fedora 28 et version ultérieure
  - b. Red Hat Enterprise Linux 6 et version ultérieure
  - c. CoreOS Container Linux 2023 et version ultérieure
  - d. Microsoft Windows 7 SP1
  - e. Microsoft Windows 10 et version ultérieure
  - f. Microsoft Windows Server 2016 et version ultérieure
- ▶ 2. **Quels sont les composants requis pour configurer votre système en tant qu'hôte de virtualisation et gérer les machines virtuelles avec la console Web ? (Choisissez-en deux.)**
  - a. Le module YUM *virt*
  - b. Le groupe de paquetages *Openstack*
  - c. Le paquetage *cockpit-machines*
  - d. Le groupe de paquetages *Virtualization Platform*
  - e. Le module YUM *kvm*
  - f. Le paquetage *cockpit-virtualization*
- ▶ 3. **Quelle commande vérifie que votre système prend en charge la virtualisation ?**
  - a. grep kvm /proc/cpuinfo
  - b. valsh valider
  - c. virt-host-validate
  - d. rhv-validate
  - e. cockpit-validate
- ▶ 4. **Quels outils pouvez-vous utiliser pour démarrer et arrêter vos machines virtuelles sur un système Red Hat Enterprise Linux ? (Choisissez-en deux.)**
  - a. vmctl
  - b. libvirtd
  - c. virsh
  - d. openstack
  - e. Console Web

## ► OPEN LAB

# INSTALLATION DE RED HAT ENTERPRISE LINUX

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez créer un fichier Kickstart et effectuer une installation Kickstart sur `serverb`.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un fichier kickstart.
- Mettre le fichier kickstart à disposition du programme d'installation.
- Exécuter une installation kickstart.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab installing-review start`. Cette commande exécute un script de démarrage pour déterminer si les machines `servera` et `serverb` sont accessibles sur le réseau. Elle configure ensuite Apache sur `serverb`. Elle configure également le menu de démarrage sur `serverb` pour que l'exercice effectue une installation kickstart.

```
[student@workstation ~]$ lab installing-review start
```

Préparez un fichier kickstart sur `serverb` comme spécifié et mettez-le à disposition à l'adresse `http://serverb.lab.example.com/ks-config/kickstart.cfg`. Effectuez une installation kickstart sur `servera` en utilisant le fichier kickstart que vous avez préparé.

1. Sur `serverb`, copiez `/root/anaconda-ks.cfg` dans `/home/student/kickstart.cfg` de sorte que l'utilisateur `student` puisse l'éditer.
2. Apportez les modifications ci-après au fichier `/home/student/kickstart.cfg`.
  - Commentez la commande `reboot`.
  - Commentez la commande `repo` pour le dépôt BaseOS. Modifiez la commande `repo` de sorte que le dépôt AppStream pointe vers `http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/`. Le nom du dépôt doit être défini sur `appstream`.
  - Modifiez la commande `url` pour utiliser `http://classroom.example.com/content/rhel8.0/x86_64/dvd/` comme source d'installation.
  - Commentez la commande `network`.

- Modifiez la commande **rootpw** pour qu'elle utilise du **texte brut** et définissez le mot de passe root sur **redhat**.
- Supprimez la ligne qui utilise la commande **auth** et ajoutez la ligne **authselect select sssd** pour définir le service **sssd** en tant que source d'identité et d'authentification.
- Simplifiez la commande **services** de sorte que seuls les services **kdump** et **rhsmdcertd** soient désactivés. Ne laissez activées que **sshd**, **rngd** et **chrony**.
- Ajoutez la commande **autopart**. Les commandes **part** et **reqpart** doivent déjà être commentées.
- Simplifiez la section **%post** afin qu'elle n'exécute qu'un script pour ajouter le texte **Kickstarted on DATE** à la fin du fichier **/etc/issue**. **DATE** est une information de variable et doit être générée par le script à l'aide de la commande **date** sans option supplémentaire.
- Simplifiez la section **%package** comme suit : incluez les paquetages **@core**, **chrony**, **dracut-config-generic**, **dracut-norescue**, **firewalld**, **grub2**, **kernel**, **rsync**, **tar** et **httpd**. Veillez à ce que le paquetage **plymouth** ne soit pas installé.

3. Validez la syntaxe de **kickstart.cfg**.
4. Mettez le fichier **/home/student/kickstart.cfg** à disposition dans **http://serverb.lab.example.com/ks-config/kickstart.cfg**
5. Revenez au système **workstation** pour vérifier votre travail.

## Évaluation

Sur **workstation**, exécutez le script **lab installing-review grade** pour noter l'exercice. Redémarrez **servera** pour effectuer une installation kickstart.

```
[student@workstation ~]$ lab installing-review grade
```

Corrigez les échecs dans **kickstart.cfg** en cours de partage à partir du serveur Web **serverb** en modifiant **/var/www/html/ks-config/kickstart.cfg** directement ou en modifiant **~/kickstart.cfg** et en le copiant dans **/var/www/html/ks-config/**.

Redémarrez **servera** pour effectuer une installation kickstart. Dans le menu GRUB, sélectionnez **Kickstart Red Hat Enterprise Linux 8** et appuyez sur **Entrée**.

## Fin

Sur **workstation**, exécutez le script **lab installing-review finish** pour mettre fin à l'exercice. Ce script supprime le serveur Web installé sur **serverb** pendant l'exercice.

```
[student@workstation ~]$ lab installing-review finish
```

Réinitialisez le système **servera** pour restaurer son état par défaut.

L'atelier est maintenant terminé.

## ► SOLUTION

# INSTALLATION DE RED HAT ENTERPRISE LINUX

## LISTE DE CONTRÔLE DES PERFORMANCES

Au cours de cet atelier, vous allez créer un fichier Kickstart et effectuer une installation Kickstart sur `serverb`.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un fichier kickstart.
- Mettre le fichier kickstart à disposition du programme d'installation.
- Exécuter une installation kickstart.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez la commande `lab installing-review start`. Cette commande exécute un script de démarrage pour déterminer si les machines `servera` et `serverb` sont accessibles sur le réseau. Elle configure ensuite Apache sur `serverb`. Elle configure également le menu de démarrage sur `serverb` pour que l'exercice effectue une installation kickstart.

```
[student@workstation ~]$ lab installing-review start
```

Préparez un fichier kickstart sur `serverb` comme spécifié et mettez-le à disposition à l'adresse `http://serverb.lab.example.com/ks-config/kickstart.cfg`. Effectuez une installation kickstart sur `servera` en utilisant le fichier kickstart que vous avez préparé.

1. Sur `serverb`, copiez `/root/anaconda-ks.cfg` dans `/home/student/kickstart.cfg` de sorte que l'utilisateur `student` puisse l'éditer.
  - 1.1. Utilisez la commande `ssh` pour vous connecter à `serverb` en tant qu'utilisateur `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Copiez `/root/anaconda-ks.cfg` sur `serverb` dans un fichier appelé `/home/student/kickstart.cfg` modifiable par `student`. Utilisez la commande `sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg` pour copier le contenu de `/`

**root/anaconda-ks.cfg** dans **/home/student/kickstart.cfg**. Si **sudo** demande le mot de passe de l'utilisateur **student**, utilisez **student** comme mot de passe.

```
[student@serverb ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

2. Apportez les modifications ci-après au fichier **/home/student/kickstart.cfg**.

- Commentez la commande **reboot**.
- Commentez la commande **repo** pour le dépôt BaseOS. Modifiez la commande **repo** de sorte que le dépôt AppStream pointe vers `http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/`. Le nom du dépôt doit être défini sur `appstream`.
- Modifiez la commande **url** pour utiliser `http://classroom.example.com/content/rhel8.0/x86_64/dvd/` comme source d'installation.
- Commentez la commande **network**.
- Modifiez la commande **rootpw** pour qu'elle utilise du **texte brut** et définissez le mot de passe `root` sur `redhat`.
- Supprimez la ligne qui utilise la commande **auth** et ajoutez la ligne **authselect select sssd** pour définir le service `sssd` en tant que source d'identité et d'authentification.
- Simplifiez la commande **services** de sorte que seuls les services `kdump` et `rhsmcertd` soient désactivés. Ne laissez activées que `sshd`, `rngd` et `chrony`.
- Ajoutez la commande **autopart**. Les commandes **part** et **reqpart** doivent déjà être commentées.
- Simplifiez la section `%post` afin qu'elle n'exécute qu'un script pour ajouter le texte **Kickstarted on DATE** à la fin du fichier **/etc/issue**. **DATE** est une information de variable et doit être générée par le script à l'aide de la commande **date** sans option supplémentaire.
- Simplifiez la section **%package** comme suit : incluez les paquetages `@core`, `chrony`, `dracut-config-generic`, `dracut-norescue`, `firewalld`, `grub2`, `kernel`, `rsync`, `tar` et `httpd`. Veillez à ce que le paquetage `plymouth` ne soit pas installé.

2.1. Commentez la directive `reboot` :

```
#reboot
```

2.2. La commande **repo** est trouvée deux fois dans **kickstart.cfg**. Commentez la commande **repo** pour le dépôt BaseOS. Modifiez la commande **repo** de sorte que le dépôt AppStream pointe vers le dépôt AppStream de la classe :

```
#repo --name="koji-override-0" --baseurl=http://download-node-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.0.0-20190213.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

2.3. Modifiez la commande **url** afin de spécifier le support source d'installation HTTP utilisé en classe :

```
url --url="http://classroom.example.com/content/rhel8.0/x86_64/dvd/"
```

2.4. Commentez la commande **network** :

```
#network --bootproto=dhcp --device=link --activate
```

2.5. Définissez le mot de passe root sur **redhat**. Remplacez la ligne commençant par **rootpw** par la ligne suivante :

```
rootpw --plaintext redhat
```

2.6. Supprimez la ligne qui utilise la commande **auth** et ajoutez la ligne **authselect select sssd** pour définir le service **sssd** en tant que source d'identité et d'authentification.

```
authselect select sssd
```

2.7. Simplifiez la commande **services** afin qu'elle ressemble exactement à ce qui suit :

```
services --disabled="kdump, rhsmcertd" --enabled="sshd, rngd, chronyd"
```

2.8. Commentez les commandes **part** et **reqpart**. Ajoutez la commande **autopart** :

```
#reqpart
# Disk partitioning information
#part / --fstype="xfs" --ondisk=vda --size=8000
autopart
```

2.9. Supprimez tout le contenu de la section **%post** et de son **%end**. Ajoutez la ligne suivante : **echo "Kickstarted on \$(date)" >> /etc/issue**  
La section **%post** entière doit ressembler à ceci.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

2.10. Simplifiez la spécification des paquetages exactement comme suit :

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
```

```
tar
httpd
-plymouth
%end
```

**3.** Validez la syntaxe de **kickstart.cfg**.

- 3.1. Utilisez la commande **ksvalidator** pour rechercher les erreurs de syntaxe dans le fichier Kickstart.

```
[student@serverb ~]$ ksvalidator kickstart.cfg
```

**4.** Mettez le fichier **/home/student/kickstart.cfg** à disposition dans `http://serverb.lab.example.com/ks-config/kickstart.cfg`

- 4.1. Copiez **kickstart.cfg** dans le répertoire **/var/www/html/ks-config/**.

```
[student@serverb ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

**5.** Revenez au système workstation pour vérifier votre travail.

- 5.1. Quittez serverb.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

## Évaluation

Sur workstation, exécutez le script **lab installing-review grade** pour noter l'exercice. Redémarrez servera pour effectuer une installation kickstart.

```
[student@workstation ~]$ lab installing-review grade
```

Corrigez les échecs dans **kickstart.cfg** en cours de partage à partir du serveur Web serverb en modifiant **/var/www/html/ks-config/kickstart.cfg** directement ou en modifiant `~/kickstart.cfg` et en le copiant dans **/var/www/html/ks-config/**.

Redémarrez servera pour effectuer une installation kickstart. Dans le menu GRUB, sélectionnez **Kickstart Red Hat Enterprise Linux 8** et appuyez sur **Entrée**.

## Fin

Sur workstation, exécutez le script **lab installing-review finish** pour mettre fin à l'exercice. Ce script supprime le serveur Web installé sur serverb pendant l'exercice.

```
[student@workstation ~]$ lab installing-review finish
```

Réinitialisez le système servera pour restaurer son état par défaut.

L'atelier est maintenant terminé.

# RÉSUMÉ

---

Dans ce chapitre, vous avez appris les principes suivants :

- Le DVD binaires RHEL 8 inclut Anaconda et tous les dépôts requis pour l'installation.
- L'ISO de démarrage RHEL 8 inclut le programme d'installation Anaconda, accédant aux dépôts sur le réseau pendant l'installation.
- Le système Kickstart effectue des installations sans assistance.
- Les fichiers Kickstart peuvent être créés à l'aide du site Web Kickstart Generator ou en copiant et en modifiant **/root/anaconda-ks.cfg**.
- Le module YUM *virt* fournit les paquetages permettant à un système RHEL de devenir un hôte de virtualisation.
- Le paquetage *cockpit-machines* ajoute le menu Virtual Machines à Cockpit.



## CHAPITRE 13

# RÉVISION COMPLÈTE

### PROJET

Tâches de révision depuis *Red Hat System Administration II*

### OBJECTIFS

- Tâches de révision depuis *Red Hat System Administration II*

### SECTIONS

- Révision complète

### ATELIER

- Atelier : Correction des problèmes de démarrage et maintenance des serveurs
- Atelier : Configuration et gestion des systèmes de fichiers et du stockage
- Atelier : Configuration et gestion de la sécurité du serveur

# RÉVISION COMPLÈTE

---

## OBJECTIFS

Après avoir terminé cette section, vous devez avoir révisé et actualisé les connaissances et les compétences acquises dans *Red Hat System Administration II*.

## RÉVISION DE RED HAT SYSTEM ADMINISTRATION II

Avant de commencer la révision complète de ce cours, vous devez être familiarisé avec les rubriques abordées dans chaque chapitre.

Vous pouvez vous référer aux précédentes sections du manuel pour en savoir plus.

### Chapitre 1, Amélioration de la productivité de la ligne de commande

Exécuter les commandes plus efficacement en utilisant les fonctionnalités avancées du shell bash, des scripts shell et divers utilitaires fournis par Red Hat Enterprise Linux.

- Automatiser des séquences de commandes en écrivant un script shell simple.
- Exécuter efficacement les commandes sur des listes d'éléments dans un script ou à partir de la ligne de commande en utilisant des boucles et des conditions.
- Rechercher le texte correspondant à un motif dans les fichiers journaux et les sortie de commande à l'aide de la commande **grep** et des expressions régulières.

### Chapitre 2, Planification de tâches à venir

Planifier l'exécution automatique des tâches dans le futur.

- Configurer une commande qui s'exécute une fois dans le futur.
- Planifier des commandes à exécuter de manière répétitive à l'aide du fichier crontab d'un utilisateur.
- Planifier des commandes à exécuter de manière répétitive à l'aide des répertoires et du fichier crontab du système.
- Activer et désactiver les minuteurs systemd, et configurer un minuteur qui gère les fichiers temporaires.

### Chapitre 3, Réglage des performances du système

Améliorer les performances du système en définissant des paramètres de réglage et en ajustant la priorité d'ordonnancement des processus.

- Optimiser les performances du système en sélectionnant un profil de réglage géré par le démon tuned.
- Définir et annuler les priorités de processus spécifiques avec les commandes nice et renice.

## Chapitre 4, Contrôle de l'accès aux fichiers à l'aide des ACL

Interpréter et définir des listes de contrôle d'accès (ACL, Access Control Lists) sur les fichiers pour gérer les situations nécessitant des permissions complexes d'accès pour l'utilisateur et le groupe.

- Décrire les cas d'utilisation des ACL, identifier les fichiers pour lesquels des ACL sont définies et interpréter les effets de ces ACL.
- Définir et supprimer les ACL sur les fichiers, et définir les ACL par défaut qui sont automatiquement créées par un répertoire sur les fichiers nouvellement créés.

## Chapitre 5, Gestion de la sécurité avec SELinux

Protéger et gérer la sécurité d'un serveur à l'aide de SELinux.

- Décrire comment SELinux protège les ressources et comment sélectionner le mode d'exécution.
- Configurer le contexte SELinux d'un fichier pour contrôler la manière dont les processus interagissent avec ce fichier.
- Configurer les valeurs booléennes SELinux pour autoriser les modifications de politique d'exécution selon différents besoins d'accès.
- Examiner les messages du journal SELinux et résoudre les refus AVC SELinux.

## Chapitre 6, Gestion du stockage de base

Créer et gérer des périphériques de stockage, des partitions, des systèmes de fichiers et des espaces d'échange à partir de la ligne de commande.

- Créer des partitions de stockage, les formater avec des systèmes de fichiers et les monter pour les utiliser.
- Créer et gérer des espaces d'échange pour compléter la mémoire physique.

## Chapitre 7, Gestion des volumes logiques

Créer et gérer les volumes logiques contenant des systèmes de fichiers et des espaces d'échange à partir de la ligne de commande.

- Créer et gérer des volumes logiques à partir de périphériques de stockage, puis les formater à l'aide de systèmes de fichiers ou les préparer avec des espaces d'échange.
- Ajouter et supprimer le stockage attribué aux groupes de volumes et augmenter de manière non destructive la taille d'un volume logique formaté avec un système de fichiers.

## Chapitre 8, Mise en œuvre de fonctionnalités de stockage avancées

Gérer le stockage à l'aide du système de gestion de stockage local Stratis et utiliser les volumes VDO pour optimiser l'espace de stockage utilisé.

- Gérer plusieurs couches de stockage à l'aide de la gestion de stockage local Stratis.
- Optimiser l'utilisation de l'espace de stockage en utilisant VDO pour compresser et dédupliquer les données sur les périphériques de stockage.

## Chapitre 9, Accès au stockage rattaché au réseau

Accéder au stockage rattaché au réseau en utilisant le protocole NFS.

- Monter, utiliser et démonter une exportation NFS à partir de la ligne de commande et au démarrage.
- Configurer le service de montage automatique avec des schémas de correspondance directe et indirecte pour monter automatiquement un système de fichiers NFS à la demande et démonter ce dernier lorsqu'il n'est plus utilisé.
- Configurer un client NFS afin qu'il utilise NFSv4 à l'aide du nouvel outil **nfscconf**.

## **Chapitre 10, Contrôle du processus de démarrage**

Gérer le processus de démarrage pour contrôler les services proposés, et résoudre et corriger les problèmes.

- Décrire le processus de démarrage de Red Hat Enterprise Linux, définir la cible par défaut utilisée lors du démarrage et démarrer un système sur une cible différente de celle par défaut.
- Se connecter à un système et modifier le mot de passe root lorsque le mot de passe root actuel a été perdu.
- Réparer manuellement les problèmes de configuration ou de corruption du système de fichiers qui arrêtent le processus de démarrage.

## **Chapitre 11, Gestion de la sécurité réseau**

Contrôler les connexions réseau aux services à l'aide du pare-feu du système et des règles SELinux.

- Accepter ou refuser les connexions réseau aux services système à l'aide des règles firewalld.
- Contrôler si les services réseau peuvent utiliser des ports réseau spécifiques en gérant les étiquettes de port SELinux.

## **Chapitre 12, Installation de Red Hat Enterprise Linux**

Installer Red Hat Enterprise Linux sur des serveurs et des machines virtuelles.

- Installer Red Hat Enterprise Linux sur un serveur.
- Automatiser le processus d'installation à l'aide de Kickstart.
- Installer une machine virtuelle sur votre serveur Red Hat Enterprise Linux à l'aide de Cockpit.

## ► OPEN LAB

# CORRECTION DES PROBLÈMES DE DÉMARRAGE ET MAINTENANCE DES SERVEURS

Dans cette révision, vous allez résoudre et corriger les problèmes de démarrage et mettre à jour la cible système par défaut. Vous planifieriez également l'exécution de tâches selon un calendrier répétitif en tant qu'utilisateur normal.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Diagnostiquer les problèmes et récupérer le système en mode de secours.
- Remplacer la cible par défaut `graphical.target` par `multi-user.target`.
- Planifier des tâches récurrentes à exécuter en tant qu'utilisateur normal.

## AVANT DE COMMENCER

Copiez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes avant de procéder à la réinitialisation. Réinitialisez les systèmes `workstation`, `servera` et `serverb` maintenant.

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-compreview1 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview1 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes sur `serverb` pour accomplir la révision complète :

- Sur `workstation`, exécutez la commande `lab rhcsa-compreview1 break1`. Ce script de rupture entraîne l'échec du processus de démarrage sur `serverb`. Il fixe également un délai plus long afin que le menu GRUB2 permette l'interruption du processus de démarrage et redémarre `serverb`.

Résolvez les problèmes possibles et remédiez à l'échec de démarrage. Le correctif doit garantir que `serverb` redémarre sans intervention. Utilisez `redhat` comme mot de passe du superutilisateur lorsque nécessaire.

- Sur `workstation`, exécutez la commande `lab rhcsa-compreview1 break2`. Ce script de rupture fait basculer la cible par défaut de la cible `multi-user` vers la cible `graphical` sur `serverb`. Il fixe également un délai plus long afin que le menu GRUB2 permette l'interruption du processus de démarrage et redémarre `serverb`.

Sur **serverb**, remplacez la cible par défaut par la cible **multi-user**. Les paramètres de la cible par défaut doivent persister après le redémarrage, sans intervention manuelle.

Utilisez la commande **sudo**, en tant qu'utilisateur **student** avec le mot de passe **student**, pour exécuter des commandes avec privilèges.

- Planifiez un travail récurrent en tant qu'utilisateur **student** qui exécute le script **/home/student/backup-home.sh** toutes les heures entre 19 h et 21 h, tous les jours sauf le samedi et le dimanche.

Téléchargez le script de sauvegarde depuis <http://materials.example.com/labs/backup-home.sh>. Ce script de sauvegarde **backup-home.sh** effectue une copie du répertoire **/home/student** de **serverb** sur **servera** dans le répertoire **/home/student/serverb-backup**. Utilisez le script **backup-home.sh** pour planifier la tâche récurrente en tant qu'utilisateur **student** sur **serverb**.

- Redémarrez le système et attendez que le démarrage soit terminé avant de passer à la notation.

## Évaluation

À partir de **workstation**, exécutez le script **lab rhcsa-compreview1 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview1 grade
```

## Fin

Sur **workstation**, exécutez **lab rhcsa-compreview1 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview1 finish
```

Enregistrez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes, puis réinitialisez **workstation**, **servera** et **serverb** avant le prochain exercice.

Vous avez maintenant terminé la révision complète.

## ► SOLUTION

# CORRECTION DES PROBLÈMES DE DÉMARRAGE ET MAINTENANCE DES SERVEURS

Dans cette révision, vous allez résoudre et corriger les problèmes de démarrage et mettre à jour la cible système par défaut. Vous planifieriez également l'exécution de tâches selon un calendrier répétitif en tant qu'utilisateur normal.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Diagnostiquer les problèmes et récupérer le système en mode de secours.
- Remplacer la cible par défaut `graphical.target` par `multi-user.target`.
- Planifier des tâches récurrentes à exécuter en tant qu'utilisateur normal.

## AVANT DE COMMENCER

Copiez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes avant de procéder à la réinitialisation. Réinitialisez les systèmes `workstation`, `servera` et `serverb` maintenant.

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-compreview1 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview1 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes sur `serverb` pour accomplir la révision complète :

- Sur `workstation`, exécutez la commande `lab rhcsa-compreview1 break1`. Ce script de rupture entraîne l'échec du processus de démarrage sur `serverb`. Il fixe également un délai plus long afin que le menu GRUB2 permette l'interruption du processus de démarrage et redémarre `serverb`.

Résolvez les problèmes possibles et remédiez à l'échec de démarrage. Le correctif doit garantir que `serverb` redémarre sans intervention. Utilisez `redhat` comme mot de passe du superutilisateur lorsque nécessaire.

- Sur `workstation`, exécutez la commande `lab rhcsa-compreview1 break2`. Ce script de rupture fait basculer la cible par défaut de la cible `multi-user` vers la cible `graphical` sur `serverb`. Il fixe également un délai plus long afin que le menu GRUB2 permette l'interruption du processus de démarrage et redémarre `serverb`.

**CHAPITRE 13 |** Révision complète

Sur **serverb**, remplacez la cible par défaut par la cible **multi-user**. Les paramètres de la cible par défaut doivent persister après le redémarrage, sans intervention manuelle.

Utilisez la commande **sudo**, en tant qu'utilisateur **student** avec le mot de passe **student**, pour exécuter des commandes avec privilèges.

- Planifiez un travail récurrent en tant qu'utilisateur **student** qui exécute le script **/home/student/backup-home.sh** toutes les heures entre 19 h et 21 h, tous les jours sauf le samedi et le dimanche.

Téléchargez le script de sauvegarde depuis <http://materials.example.com/labs/backup-home.sh>. Ce script de sauvegarde **backup-home.sh** effectue une copie du répertoire **/home/student** de **serverb** sur **servera** dans le répertoire **/home/student/serverb-backup**. Utilisez le script **backup-home.sh** pour planifier la tâche récurrente en tant qu'utilisateur **student** sur **serverb**.

- Redémarrez le système et attendez que le démarrage soit terminé avant de passer à la notation.

1. Sur **workstation**, exéutez la commande **lab rhcsa-compreview1 break1**.

```
[student@workstation ~]$ lab rhcsa-compreview1 break1
```

2. Après le démarrage de **serverb**, accédez à la console et remarquez que le processus de démarrage s'est rapidement arrêté. Prenez une minute pour réfléchir à la cause possible de ce problème.
  - 2.1. Localisez l'icône de la console **serverb**, en fonction de votre environnement de formation. Ouvrez la console.
  - 2.2. D'après l'erreur affichée, il semble qu'au moins certaines parties du système soient restées fonctionnelles.
  - 2.3. Appuyez sur **Ctrl+Alt+Suppr** pour redémarrer **serverb**.  
Lorsque le menu du chargeur de démarrage s'affiche, appuyez sur n'importe quelle touche, à l'exception de la touche **Entrée**, pour interrompre le compte à rebours.
  - 2.4. Modifiez l'entrée boot-loader par défaut, en mémoire, pour vous connecter en mode de secours.  
Appuyez sur **e** pour modifier l'entrée en cours.
  - 2.5. À l'aide des touches de direction, accédez à la ligne qui commence par **linux**. Ajoutez **systemd.unit=emergency.target** à la fin de la ligne.
  - 2.6. Appuyez sur **Ctrl+x** pour démarrer le système en utilisant la configuration modifiée.
  - 2.7. Connectez-vous en mode de secours. Le mot de passe **root** est **redhat**.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@serverb ~]#
```

**CHAPITRE 13 |** Révision complète

3. Remontez le système de fichiers / en lecture/écriture. Utilisez la commande **mount -a** pour essayer de monter l'ensemble des autres systèmes de fichiers.
- 3.1. Remontez le système de fichiers / en lecture/écriture pour modifier le système de fichiers.

```
[root@serverb ~]# mount -o remount,rw /
```

- 3.2. Utilisez la commande **mount -a** pour essayer de monter l'ensemble des autres systèmes de fichiers. Notez que l'un des systèmes de fichiers ne peut pas être monté.

```
[root@serverb ~]# mount -a
mount: /FakeMount: can't find UUID=fake.
```

- 3.3. Éditez **/etc/fstab** pour corriger le problème. Supprimez ou commentez la ligne incorrecte.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=fake      /FakeMount  xfs    defaults    0 0
```

- 3.4. Mettez à jour **systemd** pour que le système enregistre la nouvelle configuration **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[ 206.828912] systemd[1]: Reloading.
```

- 3.5. Vérifiez que le fichier **/etc/fstab** est maintenant correct en tentant de monter toutes les entrées.

```
[root@serverb ~]# mount -a
```

- 3.6. Redémarrez **serverb** et attendez que le démarrage soit terminé. Le système doit maintenant démarrer normalement.

```
[root@serverb ~]# systemctl reboot
```

4. Sur **workstation**, exécutez la commande **lab rhcsa-compreview1 break2**.

```
[student@workstation ~]$ lab rhcsa-compreview1 break2
```

Attendez la fin du redémarrage avant de continuer.

5. Sur **serverb**, basculez vers la cible **multi-user**. Définissez la cible par défaut sur **multi-user**. Utilisez la commande **sudo** pour exécuter toute commande administrative requise et, s'il vous est demandé, utilisez **student** comme mot de passe.

- 5.1. À partir de **workstation**, ouvrez une session SSH sur **serverb** en tant que **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 5.2. En tant qu'utilisateur **student** sur **serverb**, déterminez la cible par défaut.

```
[student@serverb ~]$ systemctl get-default
graphical.target
```

- 5.3. Basculez vers la cible **multi-user**. Utilisez la commande **sudo** et lorsque vous y êtes invité, utilisez le mot de passe **student**.

```
[student@serverb ~]$ sudo systemctl isolate multi-user.target
[sudo] password for student: student
```

- 5.4. Définissez **serverb** de sorte qu'il utilise la cible **multi-user** comme cible par défaut.

```
[student@serverb ~]$ sudo systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/
multi-user.target.
```

- 5.5. Redémarrez **serverb** pour vérifier que la cible **multi-user** est définie comme cible par défaut.

```
[student@serverb ~]$ sudo systemctl reboot
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

- 5.6. Après le redémarrage, ouvrez une session SSH sur **serverb** en tant qu'utilisateur **student**. Vérifiez que la cible **multi-user** est définie comme cible par défaut.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ systemctl get-default
multi-user.target
```

6. Planifiez un travail récurrent en tant qu'utilisateur **student** qui exécute le script **/home/student/backup-home.sh** toutes les heures entre 19 h et 21 h, tous les jours sauf le samedi et le dimanche.
- Utilisez le script **backup-home.sh** pour planifier la tâche récurrente. Téléchargez le script de sauvegarde depuis <http://materials.example.com/labs/backup-home.sh>.

- 6.1. Sur **serverb**, téléchargez le script de sauvegarde depuis <http://materials.example.com/labs/backup-home.sh>. Utilisez **chmod** pour rendre le script de sauvegarde exécutable.

```
[student@serverb ~]$ wget http://materials.example.com/labs/backup-home.sh
...output omitted...
[student@serverb ~]$ chmod +x backup-home.sh
```

**CHAPITRE 13 |** Révision complète

6.2. Utilisez la commande **crontab -e** pour ouvrir crontab à l'aide de l'éditeur de texte par défaut.

```
[student@serverb ~]$ crontab -e
```

6.3. Modifiez le fichier et ajoutez la ligne suivante :

```
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

Enregistrez les modifications et quittez l'éditeur.

6.4. Utilisez la commande **crontab -l** pour lister les tâches récurrentes planifiées.

```
[student@serverb ~]$ crontab -l  
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

7. Redémarrez serverb et attendez que le démarrage soit terminé avant de passer à la notation.

```
[student@serverb ~]$ sudo systemctl reboot  
[sudo] password for student: student  
Connection to serverb closed by remote host.  
Connection to serverb closed.  
[student@workstation ~]$
```

## Évaluation

À partir de workstation, exécutez le script **lab rhcsa-compreview1 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview1 grade
```

## Fin

Sur workstation, exécutez **lab rhcsa-compreview1 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview1 finish
```

Enregistrez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes, puis réinitialisez workstation, servera et serverb avant le prochain exercice.

Vous avez maintenant terminé la révision complète.

## ► OPEN LAB

# CONFIGURATION ET GESTION DES SYSTÈMES DE FICHIERS ET DU STOCKAGE

Au cours de cette révision, vous allez créer un volume logique LVM, monter un système de fichiers réseau, créer une partition d'échange qui est automatiquement activée au démarrage, configurer les fichiers temporaires inutilisés afin qu'ils soient supprimés du système et utiliser des ACL pour protéger un répertoire.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un volume logique LVM.
- Monter un système de fichiers sur le réseau.
- Créer une partition d'échange qui est automatiquement activée au démarrage.
- Configurer les fichiers temporaires inutilisés afin qu'ils soient supprimés du système.
- Utiliser les ACL pour protéger un répertoire.

## AVANT DE COMMENCER

Copiez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes avant de procéder à la réinitialisation. Réinitialisez maintenant les systèmes `workstation`, `servera` et `serverb`, à moins que vous ne veniez de le faire à la fin de l'exercice précédent.

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-compreview2 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes sur `serverb` pour accomplir la révision complète.

- Configurez un nouveau volume logique d'1 Gio appelé `vol_home` dans un nouveau groupe de volumes de 2 Gio appelé `extra_storage`. Utilisez le disque non partitionné `/dev/vdb` pour créer des partitions.
- Le nouveau volume logique `vol_home` doit être formaté avec un système de fichiers XFS et monté de façon persistante sur `/home-directories`.
- Assurez-vous que le système de fichiers réseau appelé `/share` est monté de manière persistante sur `/local-share` après le redémarrage. Le serveur NFS

**CHAPITRE 13 |** Révision complète

`servera.lab.example.com` exporte le système de fichiers réseau **/share**. Le chemin d'exportation NFS est `servera.lab.example.com:/share`.

- Créez une partition de 512 Mio sur le disque `/dev/vdc` afin de l'utiliser comme espace d'échange. Cet espace d'échange doit être activé automatiquement au démarrage.
- Créez un groupe appelé **production**. Créez les utilisateurs **production1**, **production2**, **production3** et **production4**. Assurez-vous qu'ils utilisent le nouveau groupe appelé **production** en tant que groupe supplémentaire.
- Configurez votre système pour qu'il utilise un nouveau répertoire appelé **/run/volatile** pour y stocker des fichiers temporaires. Les fichiers de ce répertoire doivent faire l'objet d'un nettoyage horaire s'ils ne sont pas utilisés plus de 30 secondes. Les permissions octales pour le répertoire doivent être **0700**. Veillez à utiliser le fichier `/etc/tmpfiles.d/volatile.conf` pour configurer le nettoyage horaire des fichiers dans **/run/volatile**.
- Créez un répertoire nommé **/webcontent**. Le propriétaire et le groupe du répertoire doivent être **root**. Les membres du groupe **production** doivent disposer d'un accès en lecture et en écriture pour ce répertoire. L'utilisateur **production1** ne devrait disposer que d'un accès en lecture. Ces permissions doivent également s'appliquer à tous les nouveaux fichiers et répertoires créés dans le répertoire **/webcontent**.

## Évaluation

À partir de `workstation`, exécutez le script **lab rhcsa-compreview2 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview2 grade
```

## Fin

Sur `workstation`, exécutez **lab rhcsa-compreview2 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview2 finish
```

Vous avez maintenant terminé la révision complète.

## ► SOLUTION

# CONFIGURATION ET GESTION DES SYSTÈMES DE FICHIERS ET DU STOCKAGE

Au cours de cette révision, vous allez créer un volume logique LVM, monter un système de fichiers réseau, créer une partition d'échange qui est automatiquement activée au démarrage, configurer les fichiers temporaires inutilisés afin qu'ils soient supprimés du système et utiliser des ACL pour protéger un répertoire.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Créer un volume logique LVM.
- Monter un système de fichiers sur le réseau.
- Créer une partition d'échange qui est automatiquement activée au démarrage.
- Configurer les fichiers temporaires inutilisés afin qu'ils soient supprimés du système.
- Utiliser les ACL pour protéger un répertoire.

## AVANT DE COMMENCER

Copiez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes avant de procéder à la réinitialisation. Réinitialisez maintenant les systèmes `workstation`, `servera` et `serverb`, à moins que vous ne veniez de le faire à la fin de l'exercice précédent.

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-compreview2 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes sur `serverb` pour accomplir la révision complète.

- Configurez un nouveau volume logique d'1 Gio appelé `vol_home` dans un nouveau groupe de volumes de 2 Gio appelé `extra_storage`. Utilisez le disque non partitionné `/dev/vdb` pour créer des partitions.
- Le nouveau volume logique `vol_home` doit être formaté avec un système de fichiers XFS et monté de façon persistante sur `/home-directories`.
- Assurez-vous que le système de fichiers réseau appelé `/share` est monté de manière persistante sur `/local-share` après le redémarrage. Le serveur NFS `servera.lab.example.com` exporte le système de fichiers réseau `/share`. Le chemin d'exportation NFS est `servera.lab.example.com:/share`.

**CHAPITRE 13 |** Révision complète

- Créez une partition de 512 Mio sur le disque `/dev/vdc` afin de l'utiliser comme espace d'échange. Cet espace d'échange doit être activé automatiquement au démarrage.
- Créez un groupe appelé `production`. Créez les utilisateurs `production1`, `production2`, `production3` et `production4`. Assurez-vous qu'ils utilisent le nouveau groupe appelé `production` en tant que groupe supplémentaire.
- Configurez votre système pour qu'il utilise un nouveau répertoire appelé `/run/volatile` pour y stocker des fichiers temporaires. Les fichiers de ce répertoire doivent faire l'objet d'un nettoyage horaire s'ils ne sont pas utilisés plus de 30 secondes. Les permissions octales pour le répertoire doivent être `0700`. Veillez à utiliser le fichier `/etc/tmpfiles.d/volatile.conf` pour configurer le nettoyage horaire des fichiers dans `/run/volatile`.
- Créez un répertoire nommé `/webcontent`. Le propriétaire et le groupe du répertoire doivent être `root`. Les membres du groupe `production` doivent disposer d'un accès en lecture et en écriture pour ce répertoire. L'utilisateur `production1` ne devrait disposer que d'un accès en lecture. Ces permissions doivent également s'appliquer à tous les nouveaux fichiers et répertoires créés dans le répertoire `/webcontent`.

1. À partir de `workstation`, ouvrez une session SSH sur `serverb` en tant que `student`.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...
```

2. Basculez vers l'utilisateur `root`.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

3. Créez une partition de 2 Gio sur `/dev/vdb`.

```
[root@serverb ~]# parted /dev/vdb mklabel msdos  
[root@serverb ~]# parted /dev/vdb mkpart primary 1GiB 3GiB
```

4. Créez un volume logique appelé `vol_home` en utilisant la partition de 2 Gio que vous avez créée sur `/dev/vdb`. Nommez le groupe de volumes `extra_storage`.

4.1. Déclarez le périphérique en mode bloc `/dev/vdb1` comme volume physique.

```
[root@serverb ~]# pvcreate /dev/vdb1  
...output omitted...
```

4.2. Créez le groupe de volumes `extra_storage` avec `/dev/vdb1`.

```
[root@serverb ~]# vgcreate extra_storage /dev/vdb1  
...output omitted...
```

4.3. Créez un volume logique d'1 Gio nommé `vol_home`.

```
[root@serverb ~]# lvcreate -L 1GiB -n vol_home extra_storage  
...output omitted...
```

5. Formatez **vol\_home** avec le type de système de fichiers XFS et montez-le sur **/home-directories**.

- 5.1. Créez un répertoire nommé **/home-directories**.

```
[root@serverb ~]# mkdir /home-directories
```

- 5.2. Formatez **/dev/extra\_storage/vol\_home** avec le type de système de fichiers XFS.

```
[root@serverb ~]# mkfs -t xfs /dev/extra_storage/vol_home  
...output omitted...
```

- 5.3. Montez de façon persistante **/dev/extra\_storage/vol\_home** sur **/home-directories**. Utilisez l'UUID de la structure lors de la création de l'entrée dans **/etc/fstab**.

```
[root@serverb ~]# lsblk -o UUID /dev/extra_storage/vol_home  
UUID  
988cf149-0667-4733-abca-f80c6ec50ab6  
[root@serverb ~]# echo "UUID=988cf149-0667-4733-abca-f80c6ec50ab6 /home-directories \  
xfs defaults 0 0" >> /etc/fstab  
[root@serverb ~]# mount -a
```

6. Assurez-vous que le système de fichiers réseau appelé **/share** est monté de manière persistante sur **/local-share** après le redémarrage. Le serveur NFS **servera.lab.example.com** exporte le système de fichiers réseau **/share**. Le chemin d'exportation NFS est **servera.lab.example.com:/share**.

- 6.1. Créez le répertoire **/local-share**.

```
[root@serverb ~]# mkdir /local-share
```

- 6.2. Ajoutez l'entrée appropriée à **/etc/fstab** afin que le système de fichiers du réseau disponible dans **servera.lab.example.com:/share** soit monté de manière persistante sur **/local-share** après le redémarrage.

```
[root@serverb ~]# echo "servera.lab.example.com:/share /local-share \  
nfs rw,sync 0 0" >> /etc/fstab
```

- 6.3. Montez le système de fichiers réseau sur **/local-share** en fonction de l'entrée dans **/etc/fstab**.

```
[root@serverb ~]# mount /local-share
```

7. Créez une partition de 512 Mio sur le disque **/dev/vdc** afin de l'utiliser comme espace d'échange. Cet espace d'échange doit être activé automatiquement lors du démarrage.

- 7.1. Créez une partition de 512 Gio sur **/dev/vdc**.

```
[root@serverb ~]# parted /dev/vdc mklabel msdos
[root@serverb ~]# parted /dev/vdc mkpart primary 1MiB 513MiB
```

7.2. Créez l'espace d'échange sur /dev/vdc1.

```
[root@serverb ~]# mkswap /dev/vdc1
...output omitted...
```

7.3. Activez l'espace d'échange de sorte qu'il persiste après le redémarrage. Utilisez l'UUID de la structure lors de la création de l'entrée dans **/etc/fstab**.

```
[root@serverb ~]# lsblk -o UUID /dev/vdc1
UUID
cc18ccb6-bd29-48a5-8554-546bf3471b69
[root@serverb ~]# echo "UUID=cc18...1b69 swap \
swap defaults 0 0" >> /etc/fstab
[root@serverb ~]# swapon -a
```

8. Créez les utilisateurs **production1**, **production2**, **production3** et **production4**. Assurez-vous qu'ils utilisent le nouveau groupe appelé **production** en tant que groupe supplémentaire.

```
[root@serverb ~]# groupadd production
[root@serverb ~]# for i in 1 2 3 4; do useradd -G production production$i; done
```

9. Configurez votre système pour qu'il utilise un nouveau répertoire appelé **/run/volatile** afin d'y stocker des fichiers temporaires. Les fichiers de ce répertoire doivent faire l'objet d'un nettoyage horaire s'ils ne sont pas utilisés plus de 30 secondes. Les permissions octales pour le répertoire doivent être **0700**. Veillez à utiliser le fichier **/etc/tmpfiles.d/volatile.conf** pour configurer le nettoyage horaire des fichiers dans **/run/volatile**.

9.1. Créez un fichier nommé **/etc/tmpfiles.d/volatile.conf** avec le contenu ci-dessous.

```
d /run/volatile 0700 root root 30s
```

9.2. Utilisez la commande **systemd-tmpfiles --create** pour créer le répertoire **/run/volatile** s'il n'existe pas.

```
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/volatile.conf
```

10. Créez un répertoire nommé **/webcontent**. Le propriétaire et le propriétaire du groupe du répertoire doivent être **root**. Les membres du groupe **production** doivent disposer d'un accès en lecture et en écriture pour ce répertoire. L'utilisateur **production1** ne devrait disposer que d'un accès en lecture. Ces permissions doivent également s'appliquer à tous les nouveaux fichiers et répertoires créés dans le répertoire **/webcontent**.

10.1. Créez le répertoire **/webcontent**.

```
[root@serverb ~]# mkdir /webcontent
```

10.2. Utilisez **setfac1** pour configurer les permissions pour **/webcontent** afin que les membres du groupe **production** aient un droit d'accès en lecture et en écriture, à l'exception de l'utilisateur **production1**, à qui seule la permission de lecture doit être accordée.

```
[root@serverb ~]# setfac1 -m u:production1:rx /webcontent  
[root@serverb ~]# setfac1 -m g:production:rwx /webcontent
```

10.3. Utilisez **setfac1** pour définir les permissions par défaut pour **/webcontent** de sorte que les permissions que vous avez appliquées à l'étape précédente s'appliquent également à tous les nouveaux fichiers et répertoires créés sous le répertoire **/webcontent**.

```
[root@serverb ~]# setfac1 -m d:u:production1:rx /webcontent  
[root@serverb ~]# setfac1 -m d:g:production:rwx /webcontent
```

10.4. Quittez le shell root de l'utilisateur.

```
[root@serverb ~]# exit  
logout
```

10.5. Déconnectez-vous de serverb.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

## Évaluation

À partir de **workstation**, exécutez le script **lab rhcsa-compreview2 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview2 grade
```

## Fin

Sur **workstation**, exécutez **lab rhcsa-compreview2 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview2 finish
```

Vous avez maintenant terminé la révision complète.

## ► OPEN LAB

# CONFIGURATION ET GESTION DE LA SÉCURITÉ DU SERVEUR

Dans cette révision, vous allez configurer l'authentification par clé SSH, modifier les paramètres du pare-feu, ajuster le mode SELinux et une valeur booléenne SELinux, et résoudre les problèmes liés à SELinux.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Configurer les clés SSH pour l'authentification par clé.
- Configurer les paramètres de pare-feu.
- Ajuster le mode SELinux et les valeurs booléennes SELinux.
- Résoudre des problèmes liés à SELinux.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-comprevew3 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-comprevew3 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes pour accomplir la révision complète :

- Générez des clés SSH pour l'utilisateur `student` sur `serverb`. Ne protégez pas la clé privée avec une phrase de passe.
- Sur `servera`, configurez l'utilisateur `student` afin qu'il puisse se connecter à l'aide de l'authentification par paire de clés SSH que vous avez créée pour `student` sur `serverb`. L'utilisateur `student` sur `serverb` doit pouvoir se connecter à `servera` à l'aide de SSH sans entrer de mot de passe. Utilisez `student` comme mot de passe de l'utilisateur `student`, si nécessaire.
- Sur `servera`, activez le mode **permissive** comme mode SELinux par défaut.
- Configurez `serverb` afin qu'il monte automatiquement le répertoire personnel de l'utilisateur `production5` lorsque l'utilisateur se connecte, à l'aide du système de fichiers réseau `/home-directories/production5`. Ce système de fichiers réseau est exporté de `servera.lab.example.com`. Ajustez la valeur booléenne SELinux adéquate de sorte que l'utilisateur `production5` puisse utiliser le répertoire personnel monté avec NFS sur `serverb` après l'authentification par clé SSH. Le mot de passe de l'utilisateur `production5` est `redhat`.

**CHAPITRE 13 |** Révision complète

- Sur **serverb**, ajustez les paramètres du pare-feu afin que les connexions SSH provenant de **servera** soient rejetées.
- Sur **serverb**, étudiez et corrigez le problème avec le démon Apache HTTPD, qui est configuré pour écouter le port 30080/TCP, mais qui ne démarre pas. Ajustez les paramètres du pare-feu afin que le port 30080/TCP soit ouvert pour les connexions entrantes.

## Évaluation

À partir de **workstation**, exécutez le script **lab rhcsa-compreview3 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview3 grade
```

## Fin

Sur **workstation**, exécutez **lab rhcsa-compreview3 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview3 finish
```

Enregistrez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes, puis réinitialisez **workstation**, **servera** et **serverb**.

Vous avez maintenant terminé la révision complète.

## ► SOLUTION

# CONFIGURATION ET GESTION DE LA SÉCURITÉ DU SERVEUR

Dans cette révision, vous allez configurer l'authentification par clé SSH, modifier les paramètres du pare-feu, ajuster le mode SELinux et une valeur booléenne SELinux, et résoudre les problèmes liés à SELinux.

## RÉSULTATS

Vous devez pouvoir réaliser les tâches suivantes :

- Configurer les clés SSH pour l'authentification par clé.
- Configurer les paramètres de pare-feu.
- Ajuster le mode SELinux et les valeurs booléennes SELinux.
- Résoudre des problèmes liés à SELinux.

## AVANT DE COMMENCER

Connectez-vous à `workstation` en tant qu'utilisateur `student` avec le mot de passe `student`.

Sur `workstation`, exécutez `lab rhcsa-comprevew3 start` pour commencer la révision complète. Ce script crée les fichiers nécessaires pour configurer correctement l'environnement.

```
[student@workstation ~]$ lab rhcsa-comprevew3 start
```

## INSTRUCTIONS

Effectuez les tâches suivantes pour accomplir la révision complète :

- Générez des clés SSH pour l'utilisateur `student` sur `serverb`. Ne protégez pas la clé privée avec une phrase de passe.
- Sur `servera`, configurez l'utilisateur `student` afin qu'il puisse se connecter à l'aide de l'authentification par paire de clés SSH que vous avez créée pour `student` sur `serverb`. L'utilisateur `student` sur `serverb` doit pouvoir se connecter à `servera` à l'aide de SSH sans entrer de mot de passe. Utilisez `student` comme mot de passe de l'utilisateur `student`, si nécessaire.
- Sur `servera`, activez le mode **permissive** comme mode SELinux par défaut.
- Configurez `serverb` afin qu'il monte automatiquement le répertoire personnel de l'utilisateur `production5` lorsque l'utilisateur se connecte, à l'aide du système de fichiers réseau `/home-directories/production5`. Ce système de fichiers réseau est exporté de `servera.lab.example.com`. Ajustez la valeur booléenne SELinux adéquate de sorte que l'utilisateur `production5` puisse utiliser le répertoire personnel monté avec NFS sur `serverb` après l'authentification par clé SSH. Le mot de passe de l'utilisateur `production5` est `redhat`.

**CHAPITRE 13 |** Révision complète

- Sur **serverb**, ajustez les paramètres du pare-feu afin que les connexions SSH provenant de **servera** soient rejetées.
- Sur **serverb**, étudiez et corrigez le problème avec le démon Apache HTTPD, qui est configuré pour écouter le port 30080/TCP, mais qui ne démarre pas. Ajustez les paramètres du pare-feu afin que le port 30080/TCP soit ouvert pour les connexions entrantes.

1. À partir de **workstation**, ouvrez une session SSH sur **serverb** en tant que **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...
```

2. Générez des clés SSH pour l'utilisateur **student** sur **serverb** à l'aide de la commande **ssh-keygen**. Ne protégez pas la clé privée avec une phrase de passe.

```
[student@serverb ~]$ ssh-keygen  
Generating public/private rsa key pair.  
Enter file in which to save the key (/home/student/.ssh/id_rsa): Enter  
Created directory '/home/student/.ssh'.  
Enter passphrase (empty for no passphrase): Enter  
Enter same passphrase again: Enter  
Your identification has been saved in /home/student/.ssh/id_rsa.  
Your public key has been saved in /home/student/.ssh/id_rsa.pub.  
The key fingerprint is:  
SHA256:1TPZ4TXYwiGWFExUGtRTHgfKQbF9hVuLa+VmH4vgkFY student@serverb.lab.example.com  
The key's randomart image is:  
+---[RSA 2048]---+  
| .+@B0** |  
| .=.#+B* |  
| . X.*o= |  
| . E +.+ |  
| S o + |  
| + . o = |  
| . o o + + |  
| . . . . |  
| |  
+---[SHA256]---+
```

3. Sur **servera**, configurez l'utilisateur **student** afin qu'il puisse se connecter à l'aide de l'authentification par paire de clés SSH que vous avez créée pour **student** sur **serverb**. L'utilisateur **student** sur **serverb** doit pouvoir se connecter à **servera** à l'aide de SSH sans entrer de mot de passe. Utilisez **student** comme mot de passe de l'utilisateur **student**, lorsque nécessaire.

- 3.1. Utilisez la commande **ssh-copy-id** pour transférer la clé publique de la paire de clés SSH de **student** sur **serverb** à **student** sur **servera**. Utilisez **student** comme mot de passe de l'utilisateur **student**, s'il vous est demandé.

```
[student@serverb ~]$ ssh-copy-id student@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/student/.ssh/
id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be established.
ECDSA key fingerprint is SHA256:g/fIMtVzDW TbTi1l00WC30sL6cHmro9Tf563NxmeyyE.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
student@servera's password: student

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'student@servera'"
and check to make sure that only the key(s) you wanted were added.
```

- 3.2. Utilisez la commande **ssh** pour vérifier que l'utilisateur **student** peut se connecter à **servera** à partir de **serverb** sans entrer de mot de passe.

```
[student@serverb ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

4. Sur **servera**, activez le mode **permissive** comme mode SELinux par défaut.

- 4.1. Modifiez **/etc/sysconfig/selinux** pour définir la valeur du paramètre **SELINUX** sur **permissive**. Vous pouvez utiliser la commande **sudo vi /etc/sysconfig/selinux** pour éditer le fichier de configuration en tant que superutilisateur. Si vous y êtes invité, utilisez le mot de passe **student**.

```
...output omitted...
#SELINUX=enforcing
SELINUX=permissive
...output omitted...
```

- 4.2. Utilisez la commande **sudo systemctl reboot** pour redémarrer le système en tant que superutilisateur.

```
[student@servera ~]$ sudo systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@serverb ~]$
```

5. Configurez **serverb** afin qu'il monte automatiquement le répertoire personnel de l'utilisateur **production5** lorsque l'utilisateur se connecte, à l'aide du système de fichiers réseau **/home-directories/production5**. Ce système de fichiers réseau est exporté de **servera.lab.example.com**. Ajustez la valeur booléenne SELinux adéquate de sorte que l'utilisateur **production5** puisse utiliser le répertoire personnel monté avec NFS sur **serverb** après l'authentification par clé SSH. Le mot de passe de l'utilisateur **production5** est **redhat**.

**CHAPITRE 13 |** Révision complète

- 5.1. Sur **serverb**, utilisez la commande **sudo -i** pour basculer vers le compte d'utilisateur **root**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 5.2. Installez le paquetage **autofs**.

```
[root@serverb ~]# yum install autofs  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Installed:  
    autofs-1:5.1.4-29.el8.x86_64  
  
Complete!
```

- 5.3. Créez le fichier de schéma de correspondance maître **autofs** nommé **/etc/auto.master.d/production5.autofs** avec le contenu ci-dessous.

```
/- /etc/auto.production5
```

- 5.4. Créez le fichier **/etc/auto.production5** avec le contenu suivant.

```
/localhome/production5 -rw servera.lab.example.com:/home-directories/production5
```

- 5.5. Relancez le service **autofs**.

```
[root@serverb ~]# systemctl restart autofs
```

6. Sur **servera**, vérifiez que l'utilisateur **production5** ne peut pas se connecter à **serverb** à l'aide de l'authentification par clé publique SSH. Une valeur booléenne SELinux est à l'origine de ce problème que vous résolvez en procédant aux étapes suivantes.

- 6.1. À partir de **workstation**, ouvrez une session SSH sur **servera** en tant que **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 6.2. Basculez vers l'utilisateur **production5** en utilisant **redhat** comme mot de passe.

```
[student@servera ~]$ su - production5  
Password: redhat  
[production5@servera ~]$
```

- 6.3. Utilisez la commande **ssh-keygen** pour générer les clés SSH en tant qu'utilisateur **production5**.

```
[production5@servera ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/production5/.ssh/id_rsa): Enter
Created directory '/home/production5/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/production5/.ssh/id_rsa.
Your public key has been saved in /home/production5/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:zmin1nmCt4H8LA+4FPimtdg81n17ATbInUFW3HSPxk4
production5@servera.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
|       .00.0. . |
|       ... .0 o |
|       . o o   E . |
|       . o *   +   |
|       .. .So    . |
|       . + =   . |
|       *.*+=.. |
|       Oo+***.o |
|       o.=o.=** |
+---[SHA256]-----+
```

- 6.4. Utilisez la commande **ssh-copy-id** pour transférer la clé publique de la paire de clés SSH de l'utilisateur production5 sur servera vers l'utilisateur production5 sur serverb. Utilisez redhat comme mot de passe de l'utilisateur production5, s'il vous est demandé.

```
[production5@servera ~]$ ssh-copy-id production5@serverb
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
production5/.ssh/id_rsa.pub"
The authenticity of host 'serverb (172.25.250.11)' can't be established.
ECDSA key fingerprint is SHA256:ciCkaRWF4g6eR9nSdPxQ7KL8czpViXal6BousK544TY.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
production5@serverb's password: redhat

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'production5@serverb'"
and check to make sure that only the key(s) you wanted were added.
```

- 6.5. Utilisez l'authentification par clé publique SSH au lieu de l'authentification par mot de passe pour vous connecter à serverb en tant que production5. Cette commande doit échouer.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
production5@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

7. Définissez les valeurs booléennes SELinux appropriées sur `serverb`, pour que `production5` puisse se connecter à `serverb` à l'aide de l'authentification par clé publique SSH et utiliser le répertoire personnel.
  - 7.1. Sur `serverb` en tant que `root`, définissez la valeur booléenne SELinux `use_nfs_home_dirs` sur `true`.

```
[root@serverb ~]# setsebool -P use_nfs_home_dirs true
```

- 7.2. Utilisez l'authentification par clé publique SSH au lieu de l'authentification par mot de passe pour vous connecter à `serverb` en tant que `production5`. La commande suivante doit aboutir.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
...output omitted...
[production5@serverb ~]$
```

8. Sur `serverb`, ajustez les paramètres du pare-feu afin que les connexions SSH provenant de `servera` soient rejetées. Le système `servera` utilise l'adresse IPv4 `172.25.250.10`.
  - 8.1. Utilisez la commande `firewall-cmd` pour ajouter l'adresse IPv4 de `servera` à la zone `firewalld` appelée `block`.

```
[root@serverb ~]# firewall-cmd --add-source=172.25.250.10/32 \
--zone=block --permanent
success
```

- 8.2. Utilisez la commande `firewall-cmd --reload` pour recharger les modifications dans les paramètres du pare-feu.

```
[root@serverb ~]# firewall-cmd --reload
success
```

9. Sur `serverb`, étudiez et corrigez le problème avec le démon Apache HTTPD, qui est configuré pour écouter le port `30080/TCP`, mais qui ne démarre pas. Ajustez les paramètres du pare-feu afin que le port `30080/TCP` soit ouvert pour les connexions entrantes.
  - 9.1. Utilisez la commande `systemctl` pour redémarrer le service `httpd.service`. Cette commande ne parvient pas à redémarrer le service.

```
[root@serverb ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 9.2. Utilisez la commande `systemctl status` pour rechercher la raison de l'échec du service `httpd`.

```
[root@serverb ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
    Active: failed (Result: exit-code) since Mon 2019-04-15 06:42:41 EDT; 5min ago
      Docs: man:httpd.service(8)
   Process: 27313 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
 Main PID: 27313 (code=exited, status=1/FAILURE)
    Status: "Reading configuration..."

Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission denied: AH00072: make_sock: could not bind to address [::]:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: no listening sockets available, shutting down
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: AH00015: Unable to open logs
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

Notez l'erreur de permission dans la sortie précédente, qui signifie que le démon httpd n'a pas réussi à se lier au port 30080/TCP. La politique SELinux peut être une restriction potentielle pour qu'une application se lie à un port. Appuyez sur **q** pour quitter la commande **systemctl** précédente.

9.3. Utilisez la commande **sealert** pour déterminer si une politique SELinux empêche httpd de se lier au port 30080/TCP.

```
[root@serverb ~]# sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----
SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 30080.

***** Plugin bind_ports (92.2 confidence) suggests *****

If you want to allow /usr/sbin/httpd to bind to network port 30080
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 30080
  where PORT_TYPE is one of the following: http_cache_port_t, http_port_t,
  jboss_management_port_t, jboss.messaging_port_t, ntop_port_t, puppet_port_t.
...output omitted...
```

## CHAPITRE 13 | Révision complète

Le message de journal précédent révèle que le port 30080/TCP n'a pas le contexte SELinux approprié **http\_port\_t**, ce qui oblige SELinux à empêcher httpd de se lier à ce port. Le message de journal produit également la syntaxe de la commande **semanage port** afin de vous permettre de résoudre facilement le problème.

- 9.4. Utilisez la commande **semanage port** pour définir le contexte SELinux approprié sur le port 30080/TCP pour httpd afin qu'il puisse s'y lier.

```
[root@serverb ~]# semanage port -a -t http_port_t -p tcp 30080
```

- 9.5. Utilisez la commande **systemctl** pour redémarrer httpd. Cette commande doit redémarrer correctement le service.

```
[root@serverb ~]# systemctl restart httpd
```

- 9.6. Ajoutez le port 30080/TCP à la zone firewalld par défaut appelée public.

```
[root@serverb ~]# firewall-cmd --add-port=30080/tcp --permanent  
success  
[root@serverb ~]# firewall-cmd --reload  
success
```

- 9.7. Quittez le shell de l'utilisateur root.

```
[root@serverb ~]# exit  
logout
```

- 9.8. Déconnectez-vous de serverb.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

## Évaluation

À partir de workstation, exécutez le script **lab rhcsa-compreview3 grade** pour confirmer que l'exercice est réussi. Corrigez toute erreur signalée et répétez le script tant que des erreurs persistent.

```
[student@workstation ~]$ lab rhcsa-compreview3 grade
```

## Fin

Sur workstation, exécutez **lab rhcsa-compreview3 finish** pour mettre fin à l'exercice. Ce script supprime les fichiers et les ressources créés au cours de l'exercice et permet de nettoyer l'environnement.

```
[student@workstation ~]$ lab rhcsa-compreview3 finish
```

Enregistrez tous les fichiers ou travaux que vous souhaitez conserver sur d'autres systèmes, puis réinitialisez `workstation`, `servera` et `serverb`.

Vous avez maintenant terminé la révision complète.

