

dnf



DNF signifie "Dandified Yum", et c'est un gestionnaire de paquets pour les systèmes d'exploitation Linux utilisant RPM (Red Hat Package Manager). Il est principalement utilisé pour gérer les paquets et les dépendances sur les distributions Fedora, CentOS et Red Hat Enterprise Linux.

Avec DNF, les utilisateurs peuvent installer, mettre à jour, désinstaller et rechercher des paquets sur leur système. Il peut également gérer les dépendances entre les paquets, de sorte que les utilisateurs n'ont pas à s'occuper de la résolution des dépendances manuellement.

Informations concernant les fichiers

- **Binaire** : `/usr/bin/dnf`
- **Fichiers de cache** : `/var/cache/dnf`
- **Configuration principale** : `/etc/dnf/dnf.conf`
- **Configuration des dépôts** : `/etc/yum.repos.d/`

Configurer un proxy pour tous les utilisateurs

Parfois, lorsque vous travaillez dans un environnement réseau limité, vous devez configurer un proxy pour accéder à Internet. Vous pouvez également configurer un cache en ligne, également appelé *proxy cache* qui sera utilisé pour stocker des copies des paquets demandés afin d'être réutilisés ultérieurement pas d'autres machines. Voici comment configurer un proxy pour tous les utilisateurs et pour un utilisateur spécifique sur Fedora en utilisant DNF.

Pour configurer un proxy pour tous les utilisateurs, vous devez modifier le fichier `/etc/dnf/dnf.conf`.

Ouvrez ce fichier en mode édition en utilisant la commande suivante :

```
sudo nano /etc/dnf/dnf.conf
```

Ajoutez les lignes suivantes pour configurer un proxy simple :

```
proxy=http://192.168.100.5:3128/
```

Si le proxy nécessite une identification, ajoutez également les lignes suivantes :

```
proxy=http://192.168.100.5:3128/  
proxy_username=vivek  
proxy_password=secretePasswordHere
```

Enregistrez et fermez le fichier. À partir de maintenant, tous les utilisateurs sur votre système seront en mesure d'utiliser le proxy pour accéder à Internet en utilisant DNF.

C'est tout! Vous savez maintenant comment configurer un proxy avec DNF pour tous les utilisateurs sur Fedora. En suivant les étapes décrites ci-dessus, vous pouvez facilement accéder à Internet en utilisant DNF même dans un environnement réseau limité.

Configurer un proxy pour un utilisateur spécifique

Pour configurer un proxy pour un utilisateur spécifique, vous devez ajouter les lignes suivantes dans le fichier `~/.bash_profile` ou `~/.bashrc` :

```
export http_proxy="http://vivek:secretePassword@192.168.100.5:3128"
```

Le fichier `~/.bash_profile` est utilisé pour les utilisateurs **non-root**, tandis que le fichier `~/.bashrc` est lu à chaque démarrage de la session utilisateur. Il est recommandé de redémarrer la session utilisateur après avoir effectué des modifications pour que les modifications prennent effet.

C'est tout! Vous savez maintenant comment configurer un proxy avec DNF pour un utilisateur spécifique sur Fedora. En suivant les étapes décrites ci-dessus, vous pouvez facilement accéder à Internet en utilisant DNF même dans un environnement réseau limité.

Rechercher un paquet installé

L'option "installed" spécifie à DNF de n'afficher que les paquets installés sur le système :

```
dnf list installed
```

-

Lister tous les paquets installés sur votre système qui correspondent à l'expression régulière "krb?-" :

```
dnf list installed "krb?-*"
```

L'expression régulière "krb?-*" signifie que DNF va rechercher tous les paquets dont le nom commence par "krb" suivi d'un caractère unique, puis suivi d'un tiret et de tout ce qui suit. Cela signifie que tous les paquets qui ont un nom similaire à "krb-xxx" ou "krb-yyy" seront listés par la commande.

Surveiller les mises à jour disponibles sur Fedora DNF

Pour surveiller les mises à jour disponibles sur votre système Fedora utilisant DNF, vous pouvez utiliser un script **bash** utilisant `dnf check-update` sans **droit root**.

```
~ 20:22:31 cedric24c
> dnf check-update
Copr repo for PyCharm owned by phracek 612 B/s | 3.6 kB 00:06
Docker CE Stable - x86_64 3.9 kB/s | 3.5 kB 00:00
~ 00:00:09 cedric24c
> dnf check-update
Dernière vérification de l'expiration des métadonnées effectuée il y a 0:00:17 le lun. 13 févr. 2023 00:00:08.
~ 00:00:26 cedric24c
```

Pour exécuter la commande `dnf check-update` en tant que service et mémoriser les informations de mise à jour dans un fichier, vous pouvez utiliser un script **bash** et une tâche **cron**.

Le script `check_update.sh` peut être déposé dans n'importe quel répertoire sur votre système, mais il est souvent pratique de le déposer dans un répertoire dédié aux scripts. Certains répertoires communs pour les scripts incluent `/usr/local/bin/` (répertoire des binaires locaux).

Voici un exemple de scripts que vous pouvez utiliser :

[check_update.sh](#)

```
#!/bin/bash

# Définir le chemin vers le fichier de mise à jour
update_file=/var/udpate_file

# Exécuter la commande de mise à jour et stocker la sortie dans le
fichier
dnf check-update
dnf check-update > $update_file
```

Enregistrez ce script dans un fichier, puis rendez-le exécutable en utilisant la commande suivante :

```
chmod +x /usr/local/bin/check_update.sh
```

La commande `touch` pour créer le fichier **update_file** s'il n'existe pas encore. La commande `chmod` avec l'option **644** permet à tous les utilisateurs (propriétaire, utilisateurs du groupe et autres utilisateurs) d'avoir uniquement des autorisations de lecture sur le fichier `update_file`

:

```
sudo touch /var/udpate_file
sudo chmod 644 /var/update_file
```

Ensuite, pour exécuter ce script en tant que service, vous pouvez ajouter une tâche **cron**. Pour ouvrir le fichier de configuration **cron**, utilisez la commande suivante :

```
sudo crontab -e
```

Ajoutez la ligne suivante à ce fichier pour exécuter le script tous les heures :

```
0 */1 * * * /usr/local/bin/check_update.sh
```

Cette ligne exécutera le script tous les heures. Vous pouvez ajuster l'heure en fonction de vos besoins.

Pour afficher les informations de mise à jour lors de l'ouverture d'un terminal, vous pouvez ajouter le code suivant à votre fichier de configuration bash `~/.bashrc` :

`.bashrc`

```
# Définir le chemin vers le fichier de mise à jour
update_file=/var/udpate_file

# Vérifier si le fichier de mise à jour existe
if [ -f $update_file ]; then
    # Compter le nombre de mises à jour
    updates=$(( $(wc -l < $update_file) - 1 )

    # Afficher le nombre de mises à jour en attente
    echo "Il y a $updates mise(s) à jour en attente."
fi
```

Après avoir enregistré les modifications, ouvrez un nouveau terminal pour vérifier que les informations de mise à jour s'affichent correctement.

Retrouver le script d'installation automatique à l'adresse [git.abonnel.fr scripts-bash](https://www.abonnel.fr/scripts-bash)

Executez la ligne suivante dans votre Terminal :

```
sudo bash -c "$(curl -fsSL  
https://git.abonnel.fr/cedricAbonnel/scripts-bash/raw/branch/main/inst  
all_check_update.sh)"
```

From:

<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:

<https://www.abonnel.fr/informatique/linux/commandes/dnf>

Last update: **2023/02/28 20:41**

