Copie et création de référentiels de packages dans Oracle® Solaris 11.2

Référence: E53759-02
Septembre 2014

ORACLE®

Table des matières

Liste des exemples

Utilisation de la présente documentation

- **Présentation** : décrit comment créer, copier, la mise à disposition, modifier et tenir à jour un référentiel de packages logiciels Oracle Solaris à l'aide de l'IPS (Image Packaging System).
- **Public visé** : Ce manuel s'adresse aux administrateurs système qui installent et gèrent des logiciels ou aident d'autres personnes à installer et gérer des logiciels.
- **Connaissances requises** : Expérience en Oracle Solaris et SMF (Service Management Facility), et expérience de l'administration de NFS et des serveurs Web.

Bibliothèque de la documentation des produits

Les informations de dernière minute et les problèmes connus pour ce produit sont inclus dans la bibliothèque de documentation accessible à l'adresse : http://www.oracle.com/pls/topic/lookup?ctx=E56338.

Accès aux services de support Oracle

Les clients Oracle ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info ou le site http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs si vous êtes malentendant.

Commentaires en retour

Faites part de vos commentaires sur cette documentation à la page suivante : http://www.oracle.com/goto/docfeedback.

♦ ♦ ♦ CHAPITRE 1

Référentiels de packages Image Packaging System

Les logiciels d'Oracle Solaris 11 sont répartis dans des packages IPS. Les packages IPS sont stockés dans des référentiels de packages IPS qui sont alimentés par des éditeurs IPS.

Ce guide décrit la création d'un référentiel de packages logiciels à l'aide de la fonction IPS (Image Packaging System) Oracle Solaris. Les outils IPS vous permettent de facilement copier un référentiel existant ou de créer un référentiel personnalisé pour vos propres packages ainsi que de mettre aisément à jour les packages dans le référentiel. Vous pouvez fournir une interface de fichier ou une interface HTTP pour les utilisateurs du référentiel. Ce guide décrit également comment automatiquement mettre à jour votre référentiel et sur la procédure à suivre pour cloner un référentiel, et indique Web Apache, configuration du serveur d'équilibrage de charge telles que le cache HTTPS d'accès, ainsi que la configuration.

Ce chapitre, vous obtenez les résultats suivants :

- Ce chapitre donne quelques raisons pour lesquelles vous pouvez souhaiter créer un référentiel de packages IPS local à usage interne.
- Les référentiels de connaître les meilleures pratiques en matière de la création du package
- Un référentiel de la configuration système minimale requise pour l'hébergement

Référentiels IPS locaux

Vous pouvez souhaiter disposer d'un référentiel IPS local pour les raisons suivantes :

- **Performances et sécurité.** Vous ne souhaitez pas que vos systèmes client accèdent à Internet pour récupérer les nouveaux packages de logiciels ou mettre à jour les packages existants.
- **Contrôle des modifications.** Vous souhaitez vous assurer que vous pourrez effectuer dans un an la même installation que celle que vous effectuez aujourd'hui. Vous souhaitez contrôler facilement les versions à les systèmes peuvent être mis à jour.
- **Packages personnalisés.** IPS dispenser personnalisé packages.

Pratiques recommandées pour la création et l'utilisation de référentiels de packages IPS locaux

Utilisez les meilleures pratiques suivantes pour tenir à jour la disponibilité des référentiels et réduire les risques d'erreur.

Incluez tous les contenus de toutes les mises à jour de référentiel de support (Support Repository Updates) (SRU).

Maintenez à jour les référentiels installés localement avec les mises à jour de support. Les mises à jour de sécurité et la prise en charge d'autres mises à jour importantes contiennent correctifs. Chaque mise à jour de la version mineure consiste Oracle Solaris et référentiel de packages est lancée comme OS un ensemble complet de packages. Les SRU sont lancés en tant qu', par exemple, pour qu'une dimension dispersée de fois que vous avez mise à jour des packages.

- N'ajoutez pas à votre référentiel de sous-ensemble de packages à partir d'une mise à jour de support. Ajoutez l'ensemble du contenu de la mise à jour de support à votre référentiel local.
- N'ignorez pas de mise à jour de support. Cumuler toutes les applicable en charge les mises à jour dans chaque référentiel.
- Ne supprimez pas les packages fournis par un éditeur Oracle.
- Utilisez le service de l'utilitaire de gestion des services (SMF)`svc:/application/pkg/mirror` pour mettre à jour automatiquement le référentiel Oracle maître local à partir du référentiel de support d'Oracle. Reportez-vous à “Copie automatique d'un référentiel à partir d'Internet” à la page 25 à propos des instructions.

Les utilisateurs peuvent mettre à jour pour une version antérieure à la version la plus récente dans le référentiel en spécifiant la version du package d'incorporation `entire` à installer. Reportez-vous auChapitre 4, “ Mise à jour ou mise à niveau d’une image Oracle Solaris ” du manuel “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ”.

Effectuez une vérification chaque fois que vous mettez à jour le référentiel.

Utilisez la commande `pkgrepo verify` à chaque fois que vous modifiez les valeurs de propriété le contenu du référentiel. La commande `pkgrepo verify`vérifie que les attributs suivants du contenu du référentiel sont correctement renseignés :

- Fichier des totaux de contrôle.
- Droits d'accès aux fichiers. Les répertoires et les fichiers du référentiel et le chemin menant vers le référentiel sont vérifiés pour s'assurer que l'utilisateur peut consulter le contenu `pkg5srv` du référentiel.
- Manifeste du package des droits d'accès.
- Contenu du manifeste de package.
- Les signatures des packages.

Créez des référentiels dans un emplacement partagé.

Un emplacement partagé est un emplacement qui ne se trouve dans aucun environnement amorçable (BE). Des exemples d'emplacements partagés sont `/var/share` et `/export`. La création d'un référentiel dans un emplacement partagé offre les avantages suivants :

- Le référentiel est d'environnements d'initialisation accéder facilement à partir d'autres.
- Lorsque vous créez un nouveau BE à mettre à niveau BE ou en clonant un ensemble existant, il n'est pas faire gaspiller de l'espace en présence de plusieurs copies d'un référentiel.
- Il n'est pas perdre du temps et de ressources d'E / S réappliquer les mises à jour du référentiel que vous avez déjà effectués dans un autre BE.

Si vous utilisez des zones non globales, les éditeurs configurés dans tous les emplacements de zones non globales doivent être accessible à partir de la zone globale, même si cet éditeur n'est pas configuré dans la zone globale.

Créez chaque référentiel dans son propre système de fichiers ZFS.

L'utilisation d'un système de fichiers ZFS distinct présente les avantages suivants :

- Amélioration des performances
- Définition de caractéristiques de systèmes de fichiers distinctes Par exemple, définissez `atime` sur `off` pour de meilleures performances lors de la mise à jour le référentiel. La propriété `atime` détermine si l'heure d'accès aux fichiers est mise à jour lorsque les fichiers sont lus. La désactivation de cette propriété évite de produire du trafic d'écriture lors de la lecture des fichiers.
- Gérer d'utilisation des ressources. Pour spécifier un référentiel quota de disque approprié pour vous assurer que chaque jeu de données ne consomment pas les mises à jour du référentiel de grande taille la totalité de l'espace dans le pool. Cette meilleure pratique est particulièrement important si vous êtes en train d'effectuer automatiquement les mises à jour comme décrit dans “Copie automatique d'un référentiel à partir d'Internet” à la page 25.
- Création d'instantané.

Créez un instantané chaque fois que vous mettez à jour le référentiel

Créer un instantané du système de fichiers du référentiel a chaque mise à jour le référentiel afin de bénéficier des avantages suivants :

- Revenir à une version antérieure à partir d'un instantané du référentiel.
- Mettre à jour le référentiel utilisateur à partir d'un instantané afin de minimiser les interruptions.

Fournissez une haute disponibilité.

- Tenir à jour du référentiel consiste à copier à différents emplacements. Reportez-vous à “Gestion de plusieurs référentiels locaux identiques” à la page 39.
- Configurer votre serveur Web pour la mise en mémoire cache, un équilibrage de charge et l'utilisation de plusieurs référentiels. Reportez-vous au Chapitre 5, Exécution du serveur de dépôt derrière un proxy Web.

Sécurisez vos référentiels locaux.

Reportez-vous à “Configuration d'un accès HTTPS à un référentiel” à la page 60 pour obtenir des instructions.

Configuration requise

Le système qui héberge le référentiel de packages IPS peut être un système x86 ou un système SPARC.

Système d’exploitation

Les serveurs de référentiel exécutant Oracle Solaris 11 11/11 prennent en charge tous les packages de mise à jour Oracle Solaris 11.

Espace disque

Pour héberger une copie du référentiel de la version Oracle Solaris 11.2, le serveur du référentiel doit disposer de 16 GO d'espace libre.

La pratique recommandée consiste à mettre à jour les référentiels installés localement avec toutes les mises à jour de support, prévoyez d'utiliser 10-15 GO d'espace disque supplémentaire chaque année pour les mises à jour. Les logiciels supplémentaires, tels que ou Oracle Solaris Studio, bien entendu Oracle Solaris Cluster requiert de l'espace supplémentaire dans le référentiel de packages.

Si un système héberge plus d'un référentiel IPS, faites de chaque référentiel un système de fichiers ZFS distinct afin de pouvoir restaurer et récupérer chaque référentiel séparément.

Privilèges de gestion du référentiel

Utilisez l'une des méthodes suivantes pour obtenir le privilège dont vous avez besoin pour créer et configurer des référentiels de packages. Reportez-vous à “ Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 ” pour plus d'informations sur les profils et les rôles, y compris les procédures permettant de déterminer le profil ou rôle qui vous intéresse.

Profils de droits

Utilisez la commande `profiles` pour répertorier les profils de droits d'accès qui vous sont affectés. Les profils suivants sont utile pour maintenir local des référentiels de packages :

Gestion de fichiers ZFS

Ce profil de droits d'accès vous permet d'exécuter la commande `zfs`.

Installation des logiciels

Ce profil de droits d'accès vous permet d'exécuter la commande `pkg`.

Gestion de services

Ce profil de droits d'accès vous permet d'exécuter des commandes SMF, telles que `svccfg`.

Rôles

Utilisez la commande `roles` pour répertorier les rôles qui vous sont affectés. Si vous disposez du rôle `root`, vous pouvez utiliser la commande `su` avec le mot de passe `root` pour prendre le rôle `root`.

Commande sudo

En fonction de la stratégie de sécurité de votre site, vous pouvez être en mesure d'utiliser la commande `sudo` avec votre mot de passe utilisateur pour exécuter une commande privilégiée.

♦ ♦ ♦ CHAPITRE 2

Copie de référentiels de packages IPS

Ce chapitre décrit deux méthodes de création d'une copie du référentiel de packages IPS Oracle Solaris Vous pouvez utiliser les fichiers de référentiel : à partir d'un média ou à partir d'un site de téléchargement Oracle Solaris, ou vous pouvez récupérer le référentiel du contenu sur Internet manuellement ou automatiquement. Dans tous les cas, créez d'abord un système de fichiers ZFS distinct dans un emplacement partagé pour votre référentiel de packages local. Une fois le référentiel instantané est créé, vérification et le référentiel.

Il contient également des informations relatives à l'ensemble du rapport de performances et la copie de référentiels de dépannage.

Considérations relatives aux peformances pour la copie de référentiels

Si vous téléchargez des fichiers de référentiel à partir du site de téléchargement, Oracle Solaris ou si vous utilisez la commande `pkgrecv` montrée dans “Copie d'un référentiel à partir d'Internet” à la page 23 pour récupérer le contenu du référentiel à partir d'un site Internet, considérez vos options de configuration suivantes pour améliorer les performances de transfert :

- Assurez-vous que votre capacité du pool de stockage ZFS est inférieure à 80%. Vous pouvez afficher la capacité du pool à l'aide de la commande `zpool list`.
- Si vous utilisez un proxy, cochez les performances du proxy.
- Fermez les applications qui utilisent une grande quantité de mémoire.
- Assurez-vous une quantité d'espace libre appropriée est disponible dans le répertoire temporaire. Au cours des opérations, la commande `pkgrecv` utilise le répertoire de stockage `$TMPDIR` en tant que mémoire temporaire. Si `TMPDIR` n'est pas définie, `pkgrecv` utilise `/var/tmp` comme mémoire temporaire. Assurez-vous que `$TMPDIR` ou `/var/tmp` ait assez d'espace libre pour la taille de l'opération `pkgrecv` que vous d'effectuez.
- Si vous utilisez la commande `pkgrecv` permettant de copier vers un référentiel important, envisagez d'utiliser l'option `--clone`. L'utilisation à l'aide de l'option `--clone` est plus rapide et consomme moins de mémoire. Reportez-vous à la section “Clonage d'un référentiel de packages IPS local” à la page 39.

- Si vous utilisez la commande `pkgrecv` pour la création ou la mise à jour d'un grand référentiel, envisagez d'utiliser un SSD pour le référentiel de destination. Vous pouvez déplacer le référentiel sous la forme d'nécessaires une fois le package exécution complète.

Dépannage des référentiels de packages locaux

Permet d'éviter des problèmes le choix entre plusieurs méthodes ou vous aider à trouver la cause des problèmes se produisant que vous pouvez rencontrer :

- Vérifiez les fichiers source du référentiel. Si vous utilisez des fichiers `.zip` pour créer votre référentiel, confirmez que les fichiers présents sur votre système sont corrects en utilisant les sommes de contrôle comme décrit à la section “Copie d'un référentiel à partir d'un fichier `zip`” à la page 19.
- Vérifiez le référentiel installé. Utilisez la commande `pkgrepo verify` pour vérifier votre référentiel installé.

  Les problèmes de droits d'accès suivants sont rapportés à l'aide de `pkgrepo verify` :

  - Droits d'accès aux fichiers. Pour éviter tout problème avec des autorisations de fichiers et répertoires pour le système de fichiers, assurez-vous que l'utilisateur `pkg5srv` est autorisé à lire le référentiel.
  - Autorisations de répertoire. Assurez-vous que tous les répertoires sur le référentiel doivent disposer des droits d'accès en exécution.

  Si la commande `pkgrepo verify` signale d'autres types d'erreur, essayez d'utiliser la commande `pkgrepo fix` afin de corriger les erreurs. Reportez-vous à la page de manuel`pkgrepo`(1) pour plus d'informations.
- Vérifiez votre origine de l'éditeur. Assurez-vous que vous définissez l'origine de chaque éditeur de manière appropriée dans chaque image. Pour mettre à jour les packages installés, d'installer des packages qui dépendent de packages installés, ou installer une zone non globale, le référentiel que vous avez définie par l'origine de l'éditeur doit contenir au moins les mêmes logiciel installé dans l'image à un emplacement où vous êtes en train de définir l'éditeur. Reportez-vous à l'étape 3 “Autorisation des utilisateurs à récupérer des packages à l'aide d'une interface de fichier” à la page 29. Reportez-vous à “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ” pour plus d'informations sur la définition des éditeurs et des problèmes d'installation de packages de dépannage.
- Vérifiez la configuration du serveur Web. Si vous configurez un serveur Web Apache pour accéder au référentiel, configuration du serveur Web afin décodent pas les barres obliques codées. Reportez-vous aux instructions dans “Paramètre de configuration Apache requis” à la page 54. Peut du décodage des barres obliques codées par le package n'est pas trouvée, une erreur d'" peuvent apparaître.
- Ne créez pas de référentiel qui est uniquement accessible à partir d'une zone non globale. Les éditeurs configurés dans tous les emplacements de zones non globales doivent être

accessible à partir de la zone globale, même si cet éditeur n'est pas configuré dans la zone globale.

Copie d'un référentiel à partir d'un fichier

Cette section décrit comment créer une copie locale du référentiel de packages Oracle Solaris à partir d'une ou plusieurs fichiers de référentiel. Les fichiers du référentiel peut se trouver sur un média ou peuvent être disponibles sur un site de téléchargement Oracle Solaris. Les fichiers du référentiel peuvent être des fichiers `zip`ou des fichiers `iso`.

▼ Copie d'un référentiel à partir d'un fichier `zip`

1. **Créez un système de fichiers ZFS pour le nouveau référentiel.**

   Créez le référentiel dans un emplacement partagé. Définissez `atime` sur `off` lors de la création du référentiel système de fichiers. Reportez-vous aux “Pratiques recommandées pour la création et l'utilisation de référentiels de packages IPS locaux” à la page 12.

   ```
   $ zfs create -o atime=off rpool/export/IPSpkgrepos
   $ zfs create rpool/export/IPSpkgrepos/Solaris
   $ zfs get atime rpool/export/IPSpkgrepos/Solaris
   NAME                              PROPERTY  VALUE  SOURCE
   rpool/export/IPSpkgrepos/Solaris  atime     off    inherited from rpool/export/IPSpkgrepos
   ```

2. **Obtenez les fichiers de référentiel de packages.**

   Téléchargez les fichiers `.zip` du référentiel de packages IPS à partir de l'emplacement d'où vous avez téléchargé l'image d'installation du système ou identifiez le DVD du référentiel dans le paquet de médias. En même temps que les fichiers `.zip`, téléchargez le script `install-repo.ksh`, ainsi que les fichiers `.txt` (et de la somme de contrôle et `README`).

   ```
   $ ls
   install-repo.ksh           sol-11_2-ga-repo-3of4.zip
   README-zipped-repo.txt     sol-11_2-ga-repo-4of4.zip
   sol-11_2-ga-repo-1of4.zip  sol-11_2-ga-repo.txt
   sol-11_2-ga-repo-2of4.zip
   ```

3. **Assurez-vous qu'il s'agit d'un fichier exécutable.**

   ```
   $ chmod +x install-repo.ksh
   ```

4. **Exécutez le script d'installation du référentiel.**

   Le script d'installation du référentiel `install-repo.ksh` décompresse chaque fichier `.zip` de référentiel, dans le répertoire indiqué. Le script, si vous le souhaitez, effectue les tâches suivantes :

- Vérifiez les sommes de contrôle des fichiers `.zip` téléchargés. Si vous ne spécifiez pas l'option `-c` pour vérifier les sommes de contrôle, alors vérifiez les sommes de contrôle manuellement avant d'exécuter le script d'installation. Exécutez la commande `digest` et comparez la sortie avec le approprié de somme de contrôle à partir du fichier `.md5` :

  ```
  $ digest -a md5 file
  ```
- La présence d'une mise à jour le référentiel de contenu ajouter si la destination spécifiée de contenu contient déjà un référentiel.
- Vérifiez la dernière référentiel. Si vous ne spécifiez pas l'option `-v` pour vérifier le référentiel, utilisez les sous-commandes `info`, `list` et `verify` de la commande `pkgrepo` pour vérifier le référentiel à l'issue de l'exécution du script d'installation.
- Créez une image ISO pour le montage et la distribution. Si vous utilisez l'option `-I` pour créer un fichier `.iso`, le fichier `.iso` et le fichier `README` qui explique comment utiliser le fichier `.iso` sont exprimés dans le répertoire de destination spécifié.

5. Vérifiez le contenu du référentiel.

Si vous n'avez pas spécifié l'option `-v` à l'étape précédente, utilisez les sous-commandes `infolistverify` et de la commande `pkgrepo`pour vérifier que le référentiel a été copié correctement. Si la commande `pkgrepo verify` signale des erreurs, essayez d'utiliser la commande `pkgrepo fix` afin de corriger les erreurs. Reportez-vous à la page de manuel `pkgrepo`(1) pour plus d'informations.

6. Prenez un instantané du nouveau référentiel.

```
$ zfs snapshot rpool/export/IPSpkgrepos/Solaris@sol-11_2_0
```

Exemple 2-1 Création d'un référentiel à partir d'un fichier `zip`

Dans cet exemple, il n'y de référentiel jusqu'à ce que les fichiers `zip` n'aient pas été extraits. Le script peut prendre les options suivantes :

`-s`	Optionnel. Permet d'indiquer le chemin complet du répertoire dans lequel se trouvent les fichiers `.zip`. Répertoire : en cours : par défaut.
`-d`	(Obligatoire) Permet d'indiquer le chemin complet du répertoire où vous souhaitez que le référentiel.
`-i`	Optionnel. Indique les fichiers à utiliser pour remplir ce référentiel. Le répertoire source peut contenir plusieurs ensembles de fichiers. `.zip` Image disponible par défaut : Le la plus récente dans le répertoire source.
`C`	Optionnel. Compare les sommes de contrôle des sommes des fichiers `.zip` avec les sommes de contrôle des fichiers spécifiés. Si vous spécifiez

`-c` sans arguments, le fichier par défaut utilisé est le fichier `.md5` pour l'image `-i` dans l'annuaire source.

`-v` Optionnel. Vérifie la dernière référentiel.

`-i` Optionnel. Image de l'crée un référentiel ISO dans le répertoire source. Contient également un fichier journal `mkiso.log` dans le répertoire source.

`-h` Optionnel. Affiche un message d'utilisation.

```
$ ./install-repo.ksh -d /export/IPSpkgrepos/Solaris -c -v -I
Comparing checksums of downloaded files...done. Checksums match.
Uncompressing sol-11_2-ga-repo-1of4.zip...done.
Uncompressing sol-11_2-ga-repo-2of4.zip...done.
Uncompressing sol-11_2-ga-repo-3of4.zip...done.
Uncompressing sol-11_2-ga-repo-4of4.zip...done.
Repository can be found in /export/IPSpkgrepos/Solaris.
Initiating repository verification.
Building ISO image...done.
ISO image and instructions for using the ISO image are at:
/tank/downloads/sol-11_2-ga-repo.iso
/tank/downloads/README-repo-iso.txt
$ ls /export/IPSpkgrepos/Solaris
COPYRIGHT         NOTICES           pkg5.repository   publisher         README-iso.txt
```

Le référentiel sur article réparable et vérification peut prendre un certain temps, mais le référentiel de contenu le devienne "le référentiel une fois que vous avez accédé sont disponibles dans la message.

Exemple 2-2 Ajout à un référentiel existant à partir d'un fichier `zip`

Dans cet exemple, le contenu des fichiers `zip` du référentiel est ajouté au contenu les fichiers figurant dans un référentiel de packages existant.

```
$ pkgrepo -s /export/IPSpkgrepos/Solaris info
PUBLISHER PACKAGES STATUS           UPDATED
solaris   4764     online           2014-03-18T05:30:57.221021Z
$ ./install-repo.ksh -d /export/IPSpkgrepos/Solaris -c -v -I
IPS repository exists at destination /export/IPSpkgrepos/Solaris
Current version: 0.175.2.0.0.35.0
Do you want to add to this repository? (y/n) y
Comparing checksums of downloaded files...done. Checksums match.
Uncompressing sol-11_2-ga-repo-1of4.zip...done.
Uncompressing sol-11_2-ga-repo-2of4.zip...done.
Uncompressing sol-11_2-ga-repo-3of4.zip...done.
Uncompressing sol-11_2-ga-repo-4of4.zip...done.
Repository can be found in /export/IPSpkgrepos/Solaris.
Initiating repository rebuild.
Initiating repository verification.
```

```
Building ISO image...done.
ISO image and instructions for using the ISO image are at:
/tank/downloads/sol-11_2-ga-repo.iso
/tank/downloads/README-repo-iso.txt
$ pkgrepo -s /export/IPSpkgrepos/Solaris info
PUBLISHER PACKAGES STATUS           UPDATED
solaris   4768     online           2014-06-02T18:11:55.640930Z
```

▼ Copie d'un référentiel à partir d'un fichier iso

1. **Créez un système de fichiers ZFS pour le nouveau référentiel.**

   Créez le référentiel dans un emplacement partagé. Définissez `atime` sur `off` lors de la création du référentiel système de fichiers. Reportez-vous aux "Pratiques recommandées pour la création et l'utilisation de référentiels de packages IPS locaux" à la page 12.

   ```
   $ zfs create -o atime=off rpool/export/IPSpkgrepos
   $ zfs create rpool/export/IPSpkgrepos/Solaris
   $ zfs get atime rpool/export/IPSpkgrepos/Solaris
   NAME                              PROPERTY  VALUE  SOURCE
   rpool/export/IPSpkgrepos/Solaris  atime     off    inherited from rpool/export/IPSpkgrepos
   ```

2. **Vous pouvez obtenir le package référentiel les fichiers image.**

   Créez un fichier `.iso` à partir des fichiers `.zip` du référentiel à l'aide de l'option `-I` comme décrit dans l'Exemple 2-1, "Création d'un référentiel à partir d'un fichier zip".

3. **Montez le fichier image.**

   Montez le fichier de référentiel `.iso`permettant d'accéder au contenu.

   ```
   $ mount -F hsfs /path/sol-11_2-repo.iso /mnt
   ```

   Pour éviter d'avoir à remonter l'image `.iso` chaque fois que le système redémarre, copiez les fichiers du référentiel comme décrit dans la section suivante.

4. **Copiez le référentiel de contenu vers le nouvel emplacement.**

   Pour augmenter les performances des accès au référentiel et pour éviter d'avoir à remonter l'image `.iso` chaque fois que le système redémarre, copiez les fichiers du référentiel de `/mnt/repo/` vers un système de fichiers ZFS. Vous pouvez effectuer cette copie à l'aide de la commande `rsync` ou à l'aide de la commande `tar`.

   - **Utilisez la commande `rsync`.**

     Si vous utilisez la commande `rsync`, n'oubliez pas d'indiquer `/mnt/repo/` (y compris la barre oblique de fin) et non `/mnt/repo` pour copier les fichiers et sous-répertoires dans le répertoire `repo`. Reportez-vous à la page de manuel `rsync`(1).

```
$ rsync -aP /mnt/repo/ /export/IPSpkgrepos/Solaris
```

- **Utilisez la commande `tar`.**

  L'utilisation de la commande `tar`, comme illustré dans l'exemple suivant, peut être un moyen plus rapide de déplacer le référentiel à partir du système de fichiers monté vers le système de fichiers ZFS du référentiel.

  ```
  $ cd /mnt/repo; tar cf - . | (cd /export/IPSpkgrepos/Solaris; tar xfp -)
  ```

5. **Démontez le fichier image.**

   Vérifiez que vous n'êtes pas encore dans le répertoire `/mnt`.

   ```
   $ umount /mnt
   ```

6. **Vérifiez que le nouveau contenu du référentiel.**

   Utilisez les sous-commandes `info`,`list` et `verify` de la commande `pkgrepo` pour vérifier que le référentiel a été copié correctement. Si la commande `pkgrepo verify` signale des erreurs, essayez d'utiliser la commande `pkgrepo fix` afin de corriger les erreurs. Reportez-vous à la page de manuel `pkgrepo`(1) pour plus d'informations.

7. **Créez un instantané du nouveau référentiel.**

   ```
   $ zfs snapshot rpool/export/IPSpkgrepos/Solaris@sol-11_2_0
   ```

Copie d'un référentiel à partir d'Internet

Cette section décrit comment créer une copie locale du référentiel de packages en copiant le référentiel à partir d'un site Internet. La première procédure décrite ci-après vous permet de données à l'origine de la commande de copie à partir de la ligne de commande. La seconde procédure décrite ci-après vous permet automatiquement à l'aide d'un service SMF copier et mettre à jour à un référentiel.

▼ Copie explicite d'un référentiel à partir d'Internet

1. **Créez un système de fichiers ZFS pour le nouveau référentiel.**

   Créez le référentiel dans un emplacement partagé. Définissez `atime` sur `off` lors de la création du référentiel système de fichiers. Reportez-vous aux “Pratiques recommandées pour la création et l'utilisation de référentiels de packages IPS locaux” à la page 12.

   ```
   $ zfs create -o atime=off rpool/export/IPSpkgrepos
   ```

```
$ zfs create rpool/export/IPSpkgrepos/Solaris
$ zfs get atime rpool/export/IPSpkgrepos/Solaris
NAME                              PROPERTY  VALUE  SOURCE
rpool/export/IPSpkgrepos/Solaris  atime     off    inherited from rpool/export/IPSpkgrepos
```

2. **Créez l'infrastructure requise du référentiel.**

   Créez l'infrastructure de référentiel `pkg`(5) requise pour pouvoir copier le référentiel. L'image méthode fichiers utilisés dans l'infrastructure de référentiel précédente. par conséquent, inclure les cette étape n'est pas nécessaire. Lorsque vous copiez le contenu du référentiel à l'aide de la commande `pkgrecv` comme décrit dans cette méthode, vous devez créer la infrastructure de référentiel de contenu, puis copier le référentiel dans cette infrastructure. Reportez-vous aux pages de manuel `pkg`(5) and `pkgrepo`(1).

   ```
   $ pkgrepo create /export/IPSpkgrepos/Solaris
   ```

3. **Copiez le référentiel de contenu vers le nouvel emplacement.**

   Vous pouvez copier le référentiel à l'aide de la commande `pkgrecv`. Cette opération peut avoir une incidence sur les performances du réseau. Le temps requis pour cette opération dépend de la bande passante du réseau et de la vitesse de connexion. Reportez-vous également à “Considérations relatives aux peformances pour la copie de référentiels” à la page 17. Si vous mettez à jour le référentiel ultérieurement, seules les modifications sont copiées, et le processus peut être bien plus rapide.

   La commande suivante extrait les dernières versions de tous les packages du référentiel de packages spécifié à la suite de l'option `-s` vers le référentiel spécifié à la suite de l'option `-d`. Si vous copiez provenant d'un site sécurisé SSL, assurez-vous que le certificat et clé requis installé, et indiquer sa clé et d'un certificat options.

   ```
   $ pkgrecv -s https://pkg.oracle.com/solaris/support -d /export/IPSpkgrepos/Solaris \
   --key /path-to-ssl_key --cert /path-to-ssl_cert '*'
   ```

   Reportez-vous à la page du manuel `pkgrecv`(1) pour plus d'informations sur les options `-m` et `--clone`. Vous ne devez pas utiliser la même option `-m latest` au plus tard à cette fin. L'utilisation d'un référentiel est trop dispersées qui peut entraîner des erreurs lors des utilisateurs tentent de mettre à jour leurs images.

4. **Vérifiez que le nouveau contenu du référentiel.**

   Utilisez les sous-commandes `info`,`list` et `verify` de la commande `pkgrepo` pour vérifier que le référentiel a été copié correctement. Si la commande `pkgrepo verify` signale des erreurs, essayez d'utiliser la commande `pkgrepo fix` afin de corriger les erreurs. Reportez-vous à la page de manuel `pkgrepo`(1) pour plus d'informations.

5. **Créez un instantané du nouveau référentiel.**

   ```
   $ zfs snapshot rpool/export/IPSpkgrepos/Solaris@sol-11_2_0
   ```

▼ Copie automatique d'un référentiel à partir d'Internet

Par défaut, le service SMF `svc:/application/pkg/mirror` effectue une opération `pkgrecv` périodique à partir de l'éditeur `solaris` avec les origines définies dans cette image à `/var/share/pkg/repositories/solaris`. Cette opération `pkgrecv` débute à 2 heures 30, un jour de chaque mois. Pour modifier ce comportement par défaut, configurez le service, comme décrit dans cette procédure.

A la fin de chaque cycle d'exécution de ce service, le référentiel les catalogues-ci sont mis à jour. Vous n'avez pas besoin de régénérer le référentiel de créer un index de recherche.

, car ce service s'exécute périodiquement à - dire le référentiel est créé et aussi stockée mis à jour. Vous évite d'avoir à utiliser la mise à jour du référentiel manuelle des instructions figurant dans ce document.

D'autres systèmes peuvent définir leur origine `solaris` automatiquement à ce clone ou à ce référentiel mis à jour automatiquement ou à un clone de ce référentiel. Un seul système doit comporter une origine Internet de l'éditeur et exécuter le service de `mirror` pour recevoir automatiquement les mises à jour.

1. **Définissez les origines de l'éditeur.**

   Par défaut le service `mirror` transfère les packages de l'éditeur `solaris` configuré dans l'image root configurée à /. Même si vous ne pouvez pas spécifier directement les origines de l'éditeur dans la configuration de service `mirror`, vous pouvez configurer la racine des images à partir de laquelle extraire ces informations. Dans la racine de l'image, utilisez `pkg set-publisher` pour configurer l'utilisation des origines de l'éditeur comme sources du transfert `pkgrecv` pour le référentiel de miroir.

   a. **(facultatif) Définissez la racine de l'image.**

      Si l'éditeur de configuration que vous voulez utiliser pour le service de miroir est différent de celui de la configuration de l'éditeur que vous voulez utiliser dans cette image, créez une image d'utilisateur dans un emplacement partagé (et non contenue dans aucun BE) et réinitialisez la valeur de la propriété `config/ref_image` dans le service `mirror` de cette nouvelle image, comme indiqué dans l'exemple suivant. Le service `service` utilisera les valeurs de la configuration de l'éditeur à partir de l'image `config/ref_image`.

      ```
      $ svccfg -s pkg/mirror:default setprop config/ref_image = /var/share/pkg/
      mirror_svc_ref_image
      $ pkg image-create /var/share/pkg/mirror_svc_ref_image
      ```

   b. **(Facultatif) Définissez les éditeurs.**

Si vous souhaitez mettre à jour votre référentiel miroir à l'aide de packages à partir d'autres éditeurs en plus de l'éditeur `solaris`, redéfinissez la valeur de la propriété `config/publishers` dans le service `mirror`, comme indiqué dans l'exemple suivant qui illustre l'ajout des éditeurs `ha-cluster` et `solarisstudio`.

```
$ svccfg -s pkg/mirror:default setprop config/publishers = solaris,ha-
cluster,solarisstudio
```

c. Définissez les origines de l'éditeur.

Parce que ce service s'exécute périodiquement, vous devez définir des origines à un référentiel par l'éditeur qui fournit des mises à jour régulières. Pour les produits, vous souhaiterez peut-être que Oracle pour définir la prise en charge par l'éditeur des origines à un référentiel pour extraire des SRU (Support Repository Updates, mises à jour). Dans l'exemple suivant, l'option `-R` n'est nécessaire que si vous configurez des éditeurs dans une autre image root. Les options `-k` et `-c` ne sont éventuellement pas nécessaires, en fonction de l'URI d'origine.

```
$ pkg -R /var/share/pkg/mirror_svc_ref_image set-publisher \
-g https://pkg.oracle.com/solaris/support/ -k ssl_key -c ssl_cert solaris
$ pkg -R /var/share/pkg/mirror_svc_ref_image set-publisher \
-g https://pkg.oracle.com/ha-cluster/support/ -k ssl_key -c ssl_cert ha-cluster
$ pkg -R /var/share/pkg/mirror_svc_ref_image set-publisher \
-g https://pkg.oracle.com/solarisstudio/support/ -k ssl_key -c ssl_cert solarisstudio
```

Utilisez l'une des commandes suivantes pour vérifier l'éditeur configuré dans l'image, procédez comme suit :

```
$ pkg -R /var/share/pkg/mirror_svc_ref_image publisher
$ pkg -R /var/share/pkg/mirror_svc_ref_image publisher solaris ha-cluster
 solarisstudio
```

2. (facultatif) Configurez les autres propriétés du service miroir.

Vous pouvez être amené à modifier les autres propriétés du service `mirror`, telles que le service de l'heure d'exécution du service ou l'emplacement du référentiel miroir.

Vous pouvez modifier la l'heure d'exécution du service de façon à correspondre au mieux qu'elle corresponde à celle des origines que vous voulez voir utilisées par la soit mis en miroir de l'éditeur à mettre à jour. Pour modifier la l'heure d'exécution du service, modifiez la valeur de la propriété `config/crontab_period`.

Pour modifier l'emplacement du référentiel miroir, modifiez la valeur de la propriété `config/repository`. Si vous modifiez l'emplacement du référentiel, que le référentiel miroir dans un emplacement partagé. Reportez-vous aux “Pratiques recommandées pour la création et l'utilisation de référentiels de packages IPS locaux” à la page 12. L'emplacement par défaut, `/var/share/pkg/repositories/solaris`, est un emplacement partagé, contenu dans aucun BE.

3. Activez le service miroir.

Utilisez la commande `svcs mirror` pour vérifier l'état du service de `mirror`.

- **Le service est désactivé et que vous souhaitez utiliser ce service.**

  a. **Si vous actualisez le service de l'instance de modifier la configuration.**

     Si vous modifiez la configuration du service `mirror`comme illustré dans les commandes `svccfg setprop` au cours des étapes précédentes, actualisez le service à ses valeurs afin d'affecter les valeurs modifiées à l'instantané en cours d'exécution. Si la sortie générée à partir de la commande `svcprop -p config mirror` n'affiche pas les valeurs que vous souhaitez inclure, assurez-vous que la sortie générée à partir de la commande `svccfg -s mirror:default listprop config` affiche les valeurs que vous souhaitez inclure. Utilisez `svcadm refresh mirror:default` ou `svccfg -s mirror:default refresh` pour affecter les valeurs modifiées dans l'instantané du service. Utilisez la commande `svcprop -p config mirror` pour confirmer que le service est configuré correctement.

  b. **Activez le service.**

     Utilisez la commande suivante pour activer le service de miroir, procédez comme suit :

     ```
     $ svcadm enable mirror:default
     ```

     Utilisez la commande `svcs mirror` pour confirmer que le service `mirror` est en ligne. Le service sera exécuté à la durée définie par la propriété `config/crontab_period`.

- **Le service est en ligne et que vous souhaitez exécuter la service maintenant.**

  Si le service est en ligne, actualisez le service pour exécuter le service immédiatement. Vous devriez voir la méthode `svc-pkg-mirror` et la commande `pkgrecv` en cours d'exécution par l'utilisateur `pkg5srv`.

- **Le service est en ligne et que vous ne souhaitez pas utiliser ce service.**

  Utilisez la commande`svcadm disable mirror` pour désactiver ce service. Vous souhaitez peut-être exécuter un seul système ce service pour tenir à jour sur un référentiel maître. Sur d'autres systèmes, vous souhaiterez peut-être que pour désactiver ce service.

- **Le service est en une opération de maintenance ou est endommagé.**

  Utilisez la commande `svcs -xvL mirror`pour obtenir plus d'informations en vue de diagnostiquer et résoudre le problème.

4. **Contrôlez le contenu du référentiel.**

   Une fois que le service `mirror` termine l'exécution, utilisez les sous-commandes `info`, `list` et `verify` de la commande `pkgrepo` pour vérifier que le référentiel est copiée ou mis à jour

correctement. Si la commande `pkgrepo verify` signale des erreurs, essayez d'utiliser la commande `pkgrepo fix` afin de corriger les erreurs. Reportez-vous à la page de manuel `pkgrepo`(1) pour plus d'informations.

Vérifiez la valeur de propriété d`config/crontab_period` pour voir le moment de l'exécution du service `mirror`. Pendant que le service est en cours d'exécution, la commande `svcs -p mirror` affiche l'état de service comme `online*` et montre les processus démarrés par ce service. Patientez jusqu'à ce que l'état du service apparaît sous comme `online` et qu'aucun processus n'est associé au service avant de vérifier le référentiel.

5. **Créez un instantané du nouveau référentiel.**

```
$ zfs snapshot rpool/VARSHARE/pkg/repositories/solaris@sol-11_2_0
```

Étapes suivantes

Vous souhaiterez peut-être ne pas les éditeurs pour effectuer une copie du contenu à partir de plusieurs en même temps. Au lieu de définir plusieurs éditeurs en une seule propriété `config/publishers`, vous pouvez créer plusieurs instances du service `pkg/mirror`. Par exemple, la propriété `config/publishers` peut être définie à `solaris` pour l'instance `default`, à `ha-cluster` pour une nouvelle instance `pkg/mirror:ha-cluster` et à `solarisstudio` pour une nouvelle instance `pkg/mirror:solarisstudio`. De même, la propriété `config/crontab_period` peut être définie différemment pour chaque instance. Vous pouvez stocker le contenu à partir de chaque éditeur dans un même référentiel, ou vous pouvez définir une valeur `config/repository` distincte pour chaque instance `pkg/mirror`.

Voir aussi

Reportez-vous à " Gestion des services système dans Oracle Solaris 11.2 " pour plus d'informations sur les commandes SMF.

♦ ♦ ♦ CHAPITRE 3

Fourniture d'un accès au référentiel

Ce chapitre décrit la procédure à effectuer pour permettre aux clients de récupérer des packages dans le référentiel local à l'aide d'une interface de fichier ou d'une interface HTTP. Un référentiel peut être configuré pour les deux types d'accès.

Autorisation des utilisateurs à récupérer des packages par le biais d'une interface de fichier

Cette section décrit comment servir les packages du référentiel local à partir d'un répertoire du réseau local.

▼ Autorisation des utilisateurs à récupérer des packages à l'aide d'une interface de fichier

1. **Configurez un partage NFS.**

   Pour permettre aux clients d'accéder au référentiel local à l'aide d'un NFS, créez et publiez un partage NFS.

   ```
   $ zfs share -o share.nfs=on rpool/export/IPSpkgrepos%ipsrepo
   ```

   Reportez-vous à la page de manuel `zfs_share`(1M) pour plus d'informations, telles que des propriétés `share.nfs` supplémentaires que vous pouvez définir.

2. **Confirmez que le partage est publié.**

   Effectuez l'un des tests suivants pour confirmer que le partage est publié :

   - **Recherchez le référentiel dans la table des systèmes de fichiers partagés.**

     ```
     $ grep repo /etc/dfs/sharetab
     /export/IPSpkgrepos     ipsrepo nfs     sec=sys,rw
     ```

- **Vérifiez si le référentiel est accessible à partir d'un système distant.**

```
$ dfshares solaris
RESOURCE                                  SERVER ACCESS  TRANSPORT
   solaris:/export/IPSpkgrepos           solaris  -       -
```

3. **Définissez l'origine de l'éditeur.**
   Pour permettre aux systèmes client d'obtenir des packages à partir du référentiel de fichiers local, définissez l'origine de l'éditeur.

   a. **Déterminez le nom de l'éditeur.**
      Utilisez la commande suivante pour déterminer les noms des éditeurs dans votre référentiel, procédez comme suit :

```
$ pkgrepo info -s /export/IPSpkgrepos/Solaris
PUBLISHER PACKAGES STATUS           UPDATED
solaris   4768     online           2014-04-02T18:11:55.640930Z
```

   b. **Vérifiez la conformité d'origine de cet éditeur.**
      Pour mettre à jour les packages installés, d'installer des packages qui dépendent de packages installés, ou installer une zone non globale, le référentiel que vous avez définie par l'origine de l'éditeur doit contenir au moins les mêmes logiciel installé dans l'image à un emplacement où vous êtes en train de définir l'éditeur. Le référentiel peut également contenir plus anciennes ou plus récente du logiciel, mais il doit contenir le même logiciel installé dans l'image.
      La commande suivante indique que le référentiel indiqué est la tâche n'a pas pour cette image approprié : origine de l'éditeur

```
$ pkg list entire
NAME (PUBLISHER)      VERSION                   IFO
entire                0.5.11-0.175.2.0.0.36.0   i--
$ pkgrepo list -Hs http://pkg.oracle.com/solaris/release
 entire@0.5.11-0.175.2.0.0.36.0
pkgrepo list: The following pattern(s) did not match any packages:
        entire@0.5.11-0.175.2.0.0.36.0
```

      La commande suivante indique que le référentiel indiqué est une origine de l'éditeur pour cette image approprié :

```
$ pkgrepo list -Hs /export/IPSpkgrepos/Solaris entire@0.5.11-0.175.2.0.0.36.0
solaris       entire       0.5.11,5.11-0.175.2.0.0.36.0:20140401T190148Z
```

   c. **Définissez l'origine de l'éditeur.**
      A l'aide de l'emplacement du référentiel et nom de l'éditeur des étapes précédentes, exécutez la commande suivante pour définir l'origine de l'éditeur, procédez comme suit :

```
$ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris
```

`-G '*'`	Supprime toutes les origines existantes de l'éditeur `solaris`.
`-M '*'`	Supprime tous les miroirs existants de l'éditeur `solaris`.
`-g`	Ajoute l'URI du référentiel local nouvellement créé comme nouvelle origine de l'éditeur `solaris`.

Reportez-vous “ Configuration des éditeurs ” du manuel “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ” pour de plus amples informations sur la configuration des éditeurs.

Si vous réinitialisez l'origine de l'éditeur dans d'autres images, effectuez le test à nouveau : d'autres images peuvent avoir un logiciel installé avec une version différente et peuvent ne pas utiliser ce référentiel. Si vous réinitialisez l'origine de l'éditeur dans d'autres systèmes, utilisez des images se trouvant sur un chemin complet pour l'argument `-g`.

Autorisation des utilisateurs à récupérer des packages par le biais d'une interface HTTP

Cette section décrit la procédure à effectuer pour servir les packages du référentiel local à l'aide du serveur de dépôt de packages.

▼ Autorisation des utilisateurs à récupérer des packages par le biais d'une interface HTTP

Le serveur de dépôt de packages `pkg.depotd`,, fournit un accès réseau aux données stockées dans un référentiel de packages. Le service `svc : / application/ pkg / server` service SMF appelle le démon `pkg.depotd`. Pour permettre aux clients d'accéder au référentiel local à l'aide d'HTTP, cette procédure montre comment configurer le service `pkg/server`. Vous pouvez configurer l'instance `par défaut`du service. Cette procédure décrit comment créer et configurer une nouvelle instance.

1. **Créez un serveur de dépôt.**

   Utilisez la sous-commande `add` pour ajouter une nouvelle instance du service `pkg/server` r nommé `solaris`.

   ```
   $ svccfg -s pkg/server add solaris
   ```

2. **Définissez le chemin vers le référentiel.**

Définissez le chemin du répertoire où cette instance du service peut trouver les données de référentiel.

```
$ svccfg -s pkg/server:solaris setprop pkg/inst_root=/export/IPSpkgrepos/Solaris
```

3. **(facultatif) Définissez le chemin vers le référentiel.**

   Numéro de port d'écoute des requêtes de package entrantes pour cette instance. Par défaut, `pkg.depotd` écoute les connexions sur le port 80. Pour modifier le port, réinitialiser la propriété `pkg/port`.

   ```
   $ svccfg -s pkg/server:solaris setprop pkg/port=81
   ```

4. **(Facultatif) Définissez d'autres propriétés.**

   Pour obtenir une liste complète des propriétés `pkg/server`, reportez-vous à la page de manuel `pkg.depotd`(1M).

   Si vous souhaitez définir plusieurs propriétés de service, utilisez la commande suivante pour modifier toutes les propriétés à la fois. N'oubliez pas de supprimer le marqueur de commentaire (#) au début de toutes les lignes que vous modifiez.

   ```
   $ svccfg -s pkg/server:solaris editprop
   ```

5. **Démarrez le service de référentiel.**

   Redémarrez le service du serveur de dépôt de packages.

   ```
   $ svcadm refresh pkg/server:solaris
   $ svcadm enable pkg/server:solaris
   ```

6. **Vérifiez que le serveur de référentiel fonctionne.**

   Pour vérifier si le serveur de référentiel fonctionne, ouvrez une fenêtre de navigateur à l'emplacement `localhost`. Par défaut, `pkg.depotd` écoute les connexions sur le port 80. Si vous avez modifié le port, ouvrez une fenêtre de navigateur à l'emplacement `localhost:`*port_number*.

7. **Définissez l'origine de l'éditeur.**

   Pour permettre aux systèmes client d'obtenir des packages à partir du référentiel de fichiers local, définissez l'origine de l'éditeur.

   a. **Déterminez le nom de l'éditeur.**

      Utilisez la commande suivante pour déterminer les noms des éditeurs dans votre référentiel, procédez comme suit :

      ```
      $ pkgrepo info -s /export/IPSpkgrepos/Solaris
      PUBLISHER PACKAGES STATUS           UPDATED
      solaris   4768     online           2014-04-02T18:11:55.640930Z
      ```

b. Vérifiez la conformité de l'origine de l'éditeur.

Pour mettre à jour les packages installés, d'installer des packages qui dépendent de packages installés, ou installer une zone non globale, le référentiel que vous avez définie par l'origine de l'éditeur doit contenir au moins les mêmes logiciel installé dans l'image à un emplacement où vous êtes en train de définir l'éditeur. Le référentiel peut également contenir plus anciennes ou plus récente du logiciel, mais il doit contenir le même logiciel installé dans l'image.

La commande suivante indique que le référentiel indiqué est la tâche n'a pas pour cette image approprié : origine de l'éditeur

```
$ pkg list entire
NAME (PUBLISHER)      VERSION                  IFO
entire                0.5.11-0.175.2.0.0.36.0   i--
$ pkgrepo list -Hs http://pkg.oracle.com/solaris/release
 entire@0.5.11-0.175.2.0.0.36.0
pkgrepo list: The following pattern(s) did not match any packages:
        entire@0.5.11-0.175.2.0.0.36.0
```

La commande suivante indique que le référentiel indiqué est une origine de l'éditeur pour cette image approprié :

```
$ pkgrepo list -Hs http://localhost:81/ entire@0.5.11-0.175.2.0.0.36.0
solaris       entire       0.5.11,5.11-0.175.2.0.0.36.0:20140401T190148Z
```

c. Définissez l'origine de l'éditeur.

Définissez l'origine de l'éditeur sur l'une des valeurs suivantes :

- L'emplacement `pkg/inst_root`.

  ```
  $ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/Solaris/ solaris
  ```

- L'emplacement `pkg/port`.

  ```
  $ pkg set-publisher -G '*' -M '*' -g http://localhost:81/ solaris
  ```

`-G '*'`	Supprime toutes les origines existantes de l'éditeur `solaris`.
`-M '*'`	Supprime tous les miroirs existants de l'éditeur `solaris`.
`-g`	Ajoute l'URI du référentiel local nouvellement créé comme nouvelle origine de l'éditeur `solaris`.

Reportez-vous “ Configuration des éditeurs ” du manuel “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ” pour de plus amples informations sur la configuration des éditeurs.

Si vous réinitialisez l'origine de l'éditeur dans d'autres images, effectuez le test à nouveau : d'autres images peuvent avoir un logiciel installé avec une version différente et peuvent ne pas utiliser ce référentiel.

Voir aussi

- "Fourniture de plusieurs référentiels par le biais de l'accès à un serveur Web" à la page 47 décrit la procédure à effectuer pour servir plusieurs référentiels à partir de plusieurs sites ou à partir d'un seul emplacement.
- Reportez-vous à la section "Référentiels multiples sous un domaine" à la page 57 pour plus d'informations sur l'exécution de plusieurs référentiels sous un seul nom de domaine avec des préfixes différents.
- "Configuration d'un accès HTTPS à un référentiel" à la page 60 décrit la procédure à suivre pour configurer l'accès au référentiel.

♦ ♦ ♦ CHAPITRE 4

Mise à jour du référentiel de packages IPS local

Ce chapitre décrit la mise à jour de packages dans un référentiel IPS, la définition ou la mise à jour des propriétés d'un référentiel et l'ajout de packages à un référentiel à partir d'une seconde source.

Mise à jour du référentiel local

Les procédures fournies dans cette section illustrent les IPS de meilleure pratique suivantes pour la mise à jour des référentiels de packages :

- Mise à jour avec tous les conserver chaque en charge les mises à jour du référentiel pour cette version. Les mises à jour de sécurité et la prise en charge d'autres mises à jour importantes contiennent correctifs.
  - N'essayez pas de choisir particulière prise en charge des correctifs permettant de s'appliquent à partir d'une mise à jour. N'ajoutez pas à votre référentiel de sous-ensemble de packages à partir d'une mise à jour de support. Ajoutez l'ensemble du contenu de la mise à jour de support à votre référentiel local. Le comportement par défaut de la commande `pkgrecv` est d'extraire toutes les versions de tous les packages.
  - N'ignorez pas de mise à jour de support. Cumuler toutes les applicable en charge les mises à jour dans chaque référentiel.

  Les utilisateurs peuvent mettre à jour pour une version antérieure à la version la plus récente dans le référentiel en spécifiant la version du package d'incorporation `entire` à installer. Reportez-vous au Chapitre 4, “ Mise à jour ou mise à niveau d’une image Oracle Solaris ” du manuel “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ”.
- Une copie du référentiel de mise à jour. Cette pratique permet de vous assurer que tous les systèmes n'ont pas accès à la référentiel alors que le référentiel est en cours de mise à jour. Créer un instantané de référentiel, avant de mettre à jour l'instantané du référentiel, effectuez les cloner la mise à jour et remplacement aux fonctions de virtualisation mises à jour du référentiel d'origine clone.

Si vous conservez plusieurs copies des référentiels de packages avec les mêmes données, suivez la procédure ci-dessous identiques pour mettre à jour l'un de ces référentiels. Reportez-vous à “Gestion de plusieurs référentiels locaux identiques” à la page 39 pour plus d'informations

sur cette procédure pour miseà jour des référentiels supplémentaires à partir du référentiel maître.

▼ Mise à jour d'un référentiel de packages IPS local.

Remarque - Vous n'avez pas besoin d'effectuer cette procédure si vous utilisez le service SMF `svc:/application/pkg/mirror` pour mettre à jour périodiquement SMF votre référentiel de service. Reportez-vous à “Copie automatique d'un référentiel à partir d'Internet” à la page 25 pour des instructions sur l'utilisation du service `mirror`.

1. **Faites un instantané ZFS du référentiel de packages.**

   Assurez-vous de disposer d'un instantané en cours du référentiel doivent être mis à jour.

   ```
   $ zfs list -t all -r rpool/export/IPSpkgrepos/Solaris
   NAME                                          USED  AVAIL  REFER  MOUNTPOINT
   rpool/export/IPSpkgrepos/Solaris             17.6G  78.4G    34K  /export/IPSpkgrepos/Solaris
   rpool/export/IPSpkgrepos/Solaris@initial         0      -  17.6G  -
   ```

   Si vous disposez déjà d'un instantané du référentiel, utilisez la commande `zfs diff` pour vérifier si l'instantané est identique à l'ensemble de données du référentiel.

   ```
   $ zfs diff rpool/export/IPSpkgrepos/Solaris@initial
   $
   ```

   Si la commande`zfs diff` ne produit aucun résultat, l'instantané est identique à son jeu de données parent, instantané et vous pouvez l'utiliser pour réaliser la mise à jour.

   Si la commande `zfs diff` génère une sortie, ou si vous ne disposez pas d'instantané du référentiel, faites un nouvel instantané, comme indiqué à Étape 6 dans “Copie explicite d'un référentiel à partir d'Internet” à la page 23. Pour utiliser cette nouvelle instantané la mise à jour.

2. **Créez un clone ZFS du référentiel de package.**

   Pour créer le référentiel instantané clone une copie du référentiel que vous pouvez mettre à jour.

   ```
   $ zfs clone rpool/export/IPSpkgrepos/Solaris@initial rpool/export/IPSpkgrepos/Solaris_tmp
   $ zfs list -r rpool/export/IPSpkgrepos/Solaris/
   NAME                                          USED  AVAIL  REFER  MOUNTPOINT
   rpool/export/IPSpkgrepos/Solaris             17.6G  78.4G    34K  /export/IPSpkgrepos/Solaris
   rpool/export/IPSpkgrepos/Solaris@initial         0      -  17.6G  -
   rpool/export/IPSpkgrepos/Solaris_tmp           76K  78.4G  17.6G  /export/IPSpkgrepos/
   Solaris_tmp
   ```

3. **Mettez à jour le clone ZFS du référentiel de packages.**

   La même façon que vous avez créé le référentiel d'origine à partir d'un emplacement HTTP ou à partir d'un fichier, vous pouvez mettre à jour votre référentiel à partir d'un emplacement HTTP ou à partir d'un fichier.

- **Mettez à jour à partir d'un fichier `zip`.**

  Reportez-vous à Exemple 2-2, “Ajout à un référentiel existant à partir d'un fichier `zip`”. Si la destination spécifiée contient déjà un référentiel de packages, le contenu du fichier `zip` est ajouté au contenu du référentiel existant.

- **Mettez à jour à partir d'un fichier ISO.**

  a. **Montez l'image ISO.**

  ```
  $ mount -F hsfs ./sol-11_2-incr-repo.iso /mnt
  ```

  b. **Copiez le fichier du contenu dans le référentiel ISO clone.**

  Utilisez la commande `rsync` ou `tar` comme indiqué à la section “Copie d'un référentiel à partir d'un fichier `iso`” à la page 22.

  ```
  $ rsync -aP /mnt/repo/ /export/IPSpkgrepos/Solaris_tmp
  ```

  c. **Démontez l'image ISO.**

- **Mettez à jour à partir d'un référentiel.**

  Copiez du contenu d'un autre référentiel vers le référentiel clone. Si vous copiez provenant d'un site sécurisé SSL, assurez-vous que le certificat et clé requis installé, et indiquer sa clé et d'un certificat options.

  ```
  $ pkgrecv -s https://pkg.oracle.com/solaris/support \
  -d /export/IPSpkgrepos/Solaris_tmp \
  --key /path-to-ssl_key --cert /path-to-ssl_cert '*'
  ```

  Reportez-vous à la page de manuel `pkgrecv`(1) pour plus d'informations sur la commande `pkgrecv`. Seuls les packages qui ont été modifiés sont mis à jour, de sorte que ce délai pour mettre à jour votre référentiel peut se révéler beaucoup moins de temps pour remplir le référentiel d'origine. Reportez-vous aux conseils relatifs aux performances dans “Considérations relatives aux peformances pour la copie de référentiels” à la page 17.

  Si l'opération `pkgrecv` est interrompue, suivez les instructions dans “Reprise d'une réception de package interrompue” à la page 38.

4. **Remplacez le référentiel fonctionnel avec le clone mis à jour.**

```
$ svcadm disable -st pkg/server:solaris
$ zfs promote rpool/export/IPSpkgrepos/Solaris_tmp
$ zfs rename rpool/export/IPSpkgrepos/Solaris rpool/export/IPSpkgrepos/Solaris_old
$ zfs rename rpool/export/IPSpkgrepos/Solaris_tmp rpool/export/IPSpkgrepos/Solaris
```

Reportez-vous à la page de manuel svcadm(1M) pour plus d'informations sur la commande svcadm.

5. **Vérifiez les informations mises à jour.**

   Utilisez la commande pkgrepo verify pour contrôler le statut de référentiel mis à jour. Reportez-vous à la page de manuel pkgrepo(1) pour plus d'informations sur les commandes pkgrepo verify et pkgrepo fix.

6. **Cataloguez les nouveaux packages et mettez à jour les index de recherche.**

   Catalogue les nouveaux packages trouvés dans le référentiel et met à jour tous les index de recherche.

   ```
   $ pkgrepo refresh -s rpool/export/IPSpkgrepos/Solaris
   ```

7. **Créez un instantané ZFS du clone récemment mis à jour du référentiel de packages.**

   ```
   $ zfs snapshot rpool/export/IPSpkgrepos/Solaris@S11U2SRU1
   ```

8. **Redémarrez le service SMF.**

   Si vous mettez à disposition le référentiel par le biais d'une interface HTTP, redémarrez le service SMF : Veillez à indiquer l'instance de service appropriée lorsque vous redémarrez le service.

   ```
   $ svcadm restart pkg/server:solaris
   ```

9. **Supprimez l'ancien référentiel.**

   Lorsque vous avez vérifié que votre référentiel mis à jour fonctionne correctement, vous pouvez supprimer l'ancien référentiel.

   ```
   $ zfs destroy rpool/export/IPSpkgrepos/Solaris_old
   ```

Reprise d'une réception de package interrompue

Si l'opération pkgrecv est interrompue, utilisez l'option c pour récupérer le contenu qui a déjà été téléchargé et reprendre le téléchargement du contenu. La valeur de *cache_dir* est fournie dans un message d'information lorsque le transfert est interrompu, comme indiqué dans l'exemple suivant :

```
PROCESS                    ITEMS       GET (MB)        SEND (MB)
...
pkgrecv: http protocol error: code: 503 reason: Service Unavailable
URL: 'https://pkg.oracle.com/solaris/support/file/file_hash
```

```
pkgrecv: Cached files were preserved in the following directory:
        /var/tmp/pkgrecv-fOGaIg
Use pkgrecv -c to resume the interrupted download.
$ pkgrecv -c /var/tmp/pkgrecv-fOGaIg \
-s https://pkg.oracle.com/solaris/support -d /export/IPSpkgrepos/Solaris_tmp \
--key /path/to/ssl_key --cert /path/to/ssl_cert '*'
Processing packages for publisher solaris ...
Retrieving and evaluating 156 package(s)...
```

Gestion de plusieurs référentiels locaux identiques

Il peut s'avérer utile de conserver plusieurs copies des référentiels de packages avec les mêmes données pour répondre aux objectifs suivants :

- Augmenter la disponibilité des fait de tenir à jour les copie dans le référentiel des noeuds différents.
- Améliorer les performances des accès au référentiel ou si vous avez de nombreux utilisateurs vos utilisateurs sont réparties sur un très.

Appliquez la procédure dans “Mise à jour d'un référentiel de packages IPS local.” à la page 36 pour mettre à jour l'un de vos référentiels de packages. Appliquez ensuite la procédure “Clonage d'un référentiel de packages IPS local” à la page 39 pour mettre à jour les référentiels supplémentaires à partir du référentiel que vous avez mis à jour en premier. Ces deux procédures sont similaires, avec une différence importante de la façon dont vous utilisez la commande `pkgrecv`. L'opération `pkgrecv` montrée dans la procédure de clonage copie les fichiers du référentiel source, avec les effets suivants :

- Cloné horodateurs des catalogues de référentiels sont exactement les mêmes que les horodatages du catalogues du référentiel source. L'équilibrage de charge est appliqué si vos référentiels, les catalogues désignent en dans toutes les doit être exactement le même les référentiels pour éviter les problèmes lorsque le programme d'équilibrage de charge clients à partir d'un seul noeud commutateurs à un autre. Reportez-vous à “Configuration de l'équilibrage de charge” à la page 58 pour plus d'informations sur l'équilibrage de charge.
- Destination des packages figurant dans le référentiel, mais pas dans le référentiel source sont supprimées du référentiel de destination. N'utilisez pas fragmentés. pour ce référentiel comme source d'une opération de clonage sauf si votre objectif est de créer une copie exacte des seules dispersées référentiel.

▼ Clonage d'un référentiel de packages IPS local

Reportez-vous à “Mise à jour d'un référentiel de packages IPS local.” à la page 36 pour plus d'informations sur ces étapes.

1. **Copiez le référentiel de destination.**

   Assurez-vous de disposer d'un instantané en cours du référentiel de destination. Clone de ce rendre un instantané ZFS.

2. **Mettez à jour la copie du référentiel de destination.**

   Utilisez la commande de `pkgrecv` pour cloner votre référentiel de packages précédemment mis à jour dans la copie locale du référentiel de destination. Reportez-vous à la page de manuel `pkgrecv`(1) pour plus d'informations sur l'opération de clonage `pkgrecv`.

   ```
   $ pkgrecv -s /net/host1/export/IPSpkgrepos/Solaris \
   -d /net/host2/export/IPSpkgrepos/Solaris_tmp --clone
   ```

3. **Remplacez le référentiel de destination fonctionnel par le clone mis à jour.**

4. **Vérifiez les informations mises à jour.**

   Utilisez la commande `pkgrepo verify` pour contrôler le statut du référentiel de destination.

5. **Créez un instantané du référentiel que vous venez de mettre à jour.**

6. **Redémarrez le service SMF.**

   Si vous mettez à disposition le référentiel par le biais d'une interface HTTP, redémarrez le service SMF : Veillez à indiquer l'instance de service appropriée lorsque vous redémarrez le service.

7. **Supprimez l'ancien référentiel.**

   Lorsque vous avez vérifié que votre référentiel mis à jour fonctionne correctement, supprimez l'ancien référentiel.

Voir aussi Si vous mettez à disposition le référentiel par le biais d'une interface HTTP, consultez la documentation associée suivante :

- “Fourniture de plusieurs référentiels par le biais de l'accès à un serveur Web” à la page 47 décrit la procédure à effectuer pour servir plusieurs référentiels à l'aide de plusieurs démons `pkg.depotd` s'exécutant sur différents ports.
- Reportez-vous à la section “Référentiels multiples sous un domaine” à la page 57 pour plus d'informations sur l'exécution de plusieurs référentiels sous un seul nom de domaine avec des préfixes différents.

Vérification et définition des propriétés du référentiel

Cette section indique comment afficher des informations sur un référentiel IPS et comment définir les propriétés d'un référentiel et d'un éditeur.

Visualisation des propriétés applicables à la totalité du référentiel

La commande suivante affiche une liste des éditeurs de packages connus du référentiel local. La colonne STATUS (ETAT) vous indique si les données des packages de l'éditeur sont actuellement en cours de traitement.

```
$ pkgrepo info -s /export/IPSpkgrepos/Solaris
PUBLISHER PACKAGES STATUS           UPDATED
solaris   4506     online           2013-07-11T23:32:46.379726Z
```

La commande suivante affiche les propriétés qui s'applique à la totalité du référentiel. Reportez-vous à la page de manuel pkgrepo(1)pour obtenir la liste complète des propriétés du référentiel et leurs descriptions, y compris les spécifications de leurs valeurs.

```
$ pkgrepo get -s /export/IPSpkgrepos/Solaris
SECTION    PROPERTY                     VALUE
publisher  prefix                       solaris
repository check-certificate-revocation False
repository signature-required-names     ()
repository trust-anchor-directory       /etc/certs/CA/
repository version                      4
```

`publisher/prefix`

Le nom de l'éditeur par défaut. Bien que des packages à partir d'un référentiel plusieurs éditeurs peut contenir qu'un seul des éditeurs, peuvent être définies en tant que l'éditeur par défaut. Cette valeur par défaut le nom de l'éditeur permet d'effectuer les opérations suivantes :

- Pour identifier un package lorsque qu'aucun éditeur n'est spécifié dans le package FMRI dans la commande `pkg`
- Pour affecter un éditeur dans un package en même temps que le package est publié dans le référentiel (à l'aide de la commande `pkgsend`(1)) et qu'aucun éditeur n'est spécifié dans le manifeste du package.

`repository/check-certificate-revocation`

Un indicateur pour la vérification du certificat. Lorsque cette valeur est définie sur `True`, la commande `pkgrepo verify` tente de déterminer si le certificat a été révoqué depuis son émission. Cette valeur doit correspondre à la valeur de la propriété de l'image`check-certificate-revocation` décrite dans “ Propriétés de l’image supplémentaires ” du manuel “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ” et dans la page de manuel`pkg`(1).

`repository/signature-required-names`

Liste de noms qui doivent être considérés comme des noms communs de certificats lors de la validation des signatures d'un package. Cette liste est utilisée par la commande `pkgrepo`

`verify`. Cette valeur doit correspondre à la valeur de l'image `signature-required-names` décrite dans " Propriétés de l'image pour les packages signés " du manuel " Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 " et dans la page de manuel `pkg`(1).

`repository/trust-anchor-directory`

Le nom de chemin absolu du répertoire contenant les ancres sécurisées pour les packages dans ce référentiel. La valeur par défaut est `/etc/certs/CA/`. Cette valeur doit correspondre à la valeur de la propriét de l'image `trust-anchor-directory` décrite dans " Propriétés de l'image supplémentaires " du manuel " Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 " et dans la page de manuel `pkg`(1).

`repository/version`

La version du format du référentiel. Cette valeur ne peut pas être définie à l'aide de la commande `pkgrepo set` affichée dans"Modificiation des valeurs des propriétés d'un référentiel" à la page 44. Vous pouvez définir cette valeur à l'aide de la commande `pkgrepo create`. Des référentiels de version 4 sont créés par défaut. Les référentiels de version 4 prennent en charge le stockage de packages pour plusieurs éditeurs.

Visualisation des propriétés de l'éditeur d'un package

La commande suivante vous permet d'afficher informations relatives à l'éditeur `solaris` dans le référentiel local. Les parenthèses indiquent que la valeur concernée peut être une liste de valeurs.

```
$ pkgrepo get -p solaris -s /export/IPSpkgrepos/Solaris
PUBLISHER SECTION    PROPERTY         VALUE
solaris   publisher  alias
solaris   publisher  prefix           solaris
solaris   repository collection-type  core
solaris   repository description      ""
solaris   repository legal-uris       ()
solaris   repository mirrors          ()
solaris   repository name             ""
solaris   repository origins          ()
solaris   repository refresh-seconds  ""
solaris   repository registration-uri ""
solaris   repository related-uris     ()
```

`publisher/prefix`

Le nom de l'éditeur spécifié dans l'option `-p`. Si aucune option `-p` n'est spécifiée, cette valeur est le nom de l'éditeur par défaut de ce référentiel, comme décrit dans la section précédente.

`repository/collection-type`

Le type des packages dans ce référentiel. Si la valeur est `core`, ce référentiel contient toutes les dépendances déclarées par les packages dans le référentiel. Si la valeur est `supplemental`, ce référentiel ne contient pas toutes les dépendances déclarées par les packages dans le référentiel.

`repository/description`

L'objet et du contenu de ce référentiel. Si ce référentiel n'est disponible à partir d'une interface HTTP, celle-ci s'affiche dans la section A propos de, dans la partie supérieure de la page principale.

`repository/legal-uris`

Une liste d'emplacements de documents fournissant des informations juridiques sur le référentiel.

`repository/mirrors`

Une liste d'emplacements de référentiels contenant les mêmes le contenu d'un package car ce référentiel.

`repository/name`

Le nom de ce référentiel. Si ce référentiel n'est disponible à partir d'une interface HTTP, celle-ci s'affiche, en haut de la page principale et dans le titre de la fenêtre.

`repository/origins`

Une liste d'emplacements de référentiels de contenu contenant et le même package car ce référentiel de métadonnées.

`repository/refresh-seconds`

Le nombre de secondes de sorte que les clients attend entre chaque recherche de les données de package mis à jour dans ce référentiel.

`repository/registration-uri`

URI représentant l'emplacement d'une ressource qui doit être utilisée pour obtenir des informations d'identification pour accéder au référentiel.

`repository/related-uris`

Une liste d'emplacements de référentiels contenant d'autres packages qui peuvent être utiles.

La commande suivante vous permet d'afficher des informations concernant la *section/property* spécifiée dans le référentiel `pkg.oracle.com`.

```
$ pkgrepo get -p solaris -s http://pkg.oracle.com/solaris/release \
repository/name repository/description
PUBLISHER SECTION    PROPERTY    VALUE
```

```
solaris   repository description This\ repository\ serves\ the\ Oracle\ Solaris\ 11\ Package\
 repository.
solaris   repository name        Oracle\ Solaris\ 11\ Package\ Repository
```

Modificiation des valeurs des propriétés d'un référentiel

“Visualisation des propriétés de l'éditeur d'un package” à la page 42 indique que le référentiel nom et la description des valeurs de propriété référentiel ne sont pas définies pour l'éditeur `solaris` dans le référentiel local. Si cette HTTP est disponible à partir d'un référentiel et que vous utilisez un type de navigateur interface permettant de visualiser le contenu de ce référentiel, vous voyez un nom par défaut et aucune description. Une fois que vous avez défini ces des valeurs, la valeur `repository/name` d'un éditeur apparaît dans la partie supérieure de la page et comme titre de la page, et la valeur `repository/description` d'un éditeur s'affiche dans la section A propos de juste en-dessous du nom. Vous devez utiliser l'option `-p` pour indiquer au moins un éditeur lorsque vous définissez ces valeurs. Si ce référentiel contient le contenu à partir de plusieurs éditeurs, vous pouvez définir différentes valeurs pour chaque éditeur, ou vous pouvez indiquer `-p all`.

```
$ pkgrepo set -p solaris -s /export/IPSpkgrepos/Solaris \
repository/description="Local copy of the Oracle Solaris 11 repository." \
repository/name="Oracle Solaris 11"
$ pkgrepo get -p solaris -s /export/IPSpkgrepos/Solaris repository/name repository/
description
PUBLISHER SECTION    PROPERTY         VALUE
solaris   repository description      Local\ copy\ of\ the\ Oracle\ Solaris\ 11\ repository.
solaris   repository name             Oracle\ Solaris\ 11
```

Personnalisation du référentiel local

Vous pouvez utiliser la commande `pkgrecv` pour ajouter des packages et leurs données d'éditeur à votre référentiel. Vous pouvez utiliser la commande `pkgrepo` pour supprimer des packages et éditeurs à partir du référentiel.

Ajout de packages à votre référentiel

Vous pouvez ajouter des éditeurs à un référentiel. Par exemple, vous pouvez conserver les packages `solaris`, `ha-cluster` et `solarisstudio` dans un même référentiel.

Si vous ajoutez des modules personnalisés, personnalisé les publier à leur tour sous un nom de l'éditeur de packages. N'effectuez pas de publication de packages personnalisés comme un

éditeur existant comme `solaris`. Si vous publiez des packages qui ne disposent pas d'un éditeur spécifié, ces packages sera ajouté à l'éditeur par défaut pour le référentiel. Test publier des packages dans un référentiel personnalisé avec la bonne éditeur par défaut. Exécutez ensuite la commande`pkgrecv` pour ajouter ces packages et des informations relatives à l'éditeur de votre production référentiel. Reportez-vous à la section " Publish the Package " du manuel " Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ".

Dans l'exemple suivant, les données de l'éditeur `isvpub` et tous les packages de l'archive de packages `ISVproducts.p5p` sont ajoutées au référentiel local. Une *archive de package* est un fichier qui contient les informations de l'éditeur et un ou plusieurs packages fournis par cet éditeur. Reportez-vous à " Deliver as a Package Archive File " du manuel " Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2 ". La plupart des opérations `pkgrepo` ne sont pas disponibles pour des archives de packages. Une archive de package contient les packages mais ne contient pas la configuration du référentiel. Toutefois, les commandes `pkgrepo list` et `pkgrepo contents` fonctionnent avec des archives de packages. La commande `pkgrepo contents` est abordée dans "Examen de packages dans votre référentiel" à la page 46.

Dans la sortie `pkg list`, l'éditeur est affiché car il n'est pas l'éditeur qui est le mieux classé dans l'ordre de recherche dans cette image.

```
$ pkgrepo -s /tmp/ISVproducts.p5p list
PUBLISHER NAME                                    O VERSION
isvpub    isvtool                                   1.1,5.11:20131120T021902Z
isvpub    isvtool                                   1.0,5.11:20131120T010105Z
```

La commande suivante `pkgrecv` extrait tous les packages du référentiel source. Pour extraire les noms de listes ou des packages à extraire ou que vous spécifiez un m modèle, autre que `'*'`, vous devez indiquer l'option `-r` afin de vous assurer que vous d'extraire tous les packages de dépendance.

```
$ pkgrecv -s /tmp/ISVproducts.p5p -d /export/IPSpkgrepos/Solaris '*'
Processing packages for publisher isvpub ...
Retrieving and evaluating 2 package(s)...
PROCESS        ITEMS      GET (MB)      SEND (MB)
Completed        2/2       0.0/0.0          0.0/0
```

Une fois que vous avez modifié le contenu d'un référentiel, actualisez le serveur et redémarrer toute instance de service de dépôt de packages configurés pour ce référentiel.

```
$ pkgrepo -s /export/IPSpkgrepos/Solaris refresh -p isvpub
Initiating repository refresh.
$ svcadm refresh pkg/server:solaris
$ svcadm restart pkg/server:solaris
```

La commande suivante `pkgrepo info` indique un package, car les deux des packages qui ont été extraits sont différentes versions du même package. La commande `pkgrepo list` répertorie les deux packages.

```
$ pkgrepo -s /export/IPSpkgrepos/Solaris info
PUBLISHER PACKAGES STATUS           UPDATED
solaris   4768     online           2014-01-02T19:19:06.983979Z
isvpub    1        online           2014-03-20T23:24:37.196773Z
$ pkgrepo -s /export/IPSpkgrepos/Solaris list -p isvpub
PUBLISHER NAME                                  O VERSION
isvpub    isvtool                                 1.1,5.11:20131120T021902Z
isvpub    isvtool                                 1.0,5.11:20131120T010105Z
```

Ajoutez le nouvel emplacement de référentiel l'éditeur `isvpub` à l'aide de la commande `pkg set-publisher`.

Si cette HTTP est disponible à partir d'un référentiel et que vous utilisez un type de navigateur interface permettant de visualiser le contenu de ce référentiel, vous pouvez visualiser ce nouveau package en spécifiant l'éditeur dans l'emplacement. Par exemple, vous pouvez spécifier `http://localhost:81/isvpub/`.

Examen de packages dans votre référentiel

En plus des commandes `pkgrepo info` et `pkgrepo list` présentées dans “Ajout de packages à votre référentiel” à la page 44, vous pouvez utiliser la commande `pkgrepo contents` afin d'examiner le contenu de packages dans votre référentiel.

Pour un package individuel, la sortie de la commande `pkgrepo contents` est identique à la sortie de la commande `pkgrepo contents`. La commande `pkgrepo contents` affiche le résultat de chaque package correspondant ne soit dans le référentiel indiqué, tandis que la commande `pkg contents` affiche uniquement le résultat dans les versions de packages correspondants qui sont installables dans cette image. Si vous indiquez l'option `-t`, la commande `pkgrepo contents`n'affiche que les actions spécifiées.

L'exemple suivant n'a pas besoin d'indiquer la version du package Etant donné qu'une seule version de ce package existe dans le référentiel indiqué. Ce package contient des actions `depend`pour fournir l'ensemble des packages obligatoires pour Oracle Solaris et l'opération de d'installation Oracle Database 12.

```
$ pkgrepo -s http://pkg.oracle.com/solaris/release/ \
contents -t depend oracle-rdbms-server-12cR1-preinstall
depend fmri=x11/library/libxi type=group
depend fmri=x11/library/libxtst type=group
depend fmri=x11/session/xauth type=group
depend fmri=compress/unzip type=require
depend fmri=developer/assembler type=require
depend fmri=developer/build/make type=require
```

Suppression de packages de votre référentiel

Ne supprimez pas les packages fournis par un éditeur Oracle. “ Ajout et mise à jour de logiciels dans Oracle Solaris 11.2 ” indique les méthodes d'installation pour les packages que vous souhaitez uniquement en évitant l'installation des packages que vous ne souhaitez pas.

Vous pouvez utiliser la commande `pkgrepo remove` pour supprimer les modules qui n'ont pas été livrés par un éditeur Oracle. Vous pouvez utiliser la commande `pkgrepo remove-publisher` pour supprimer un éditeur, ainsi que tous les paquets fournis par cet éditeur. Reportez-vous à la page de manuel `pkgrepo`(1) pour plus d'informations. Ces opérations doivent être effectuées sur une copie du référentiel, comme décrit dans “Mise à jour d'un référentiel de packages IPS local.” à la page 36.

Fourniture de plusieurs référentiels par le biais de l'accès à un serveur Web

Les procédures décrites dans cette section expliquent comment étendre les informations fournies dans “Autorisation des utilisateurs à récupérer des packages par le biais d'une interface HTTP” à la page 31 pour permettre l'utilisation de plusieurs référentiels.

Les méthodes citées ci-après sont de deux méthodes pour servir plusieurs référentiels de packages IPS à l'aide de HTTP afin d'obtenir un accès. Pour les deux méthodes, commencez par créer instances supplémentaires du service `pkg/server` avec des chemins d'accès uniques au référentiels.

- Plusieurs emplacements. En consultant accès des utilisateurs à chacune des pages lors de référentiel des emplacements distincts.
- Seul emplacement. Les utilisateurs accèdent à tous les référentiels à partir d'un seul emplacement.

Outre l'accès à plusieurs référentiels, rappelez-vous qu'un seul référentiel peut fournir des packages à partir de plusieurs éditeurs, comme illustré dans la section “Ajout de packages à votre référentiel” à la page 44.

▼ Fourniture de plusieurs référentiels à partir d'emplacements distincts

Dans cet exemple, le référentiel `SolarisStudio` vient s'ajouter au référentiel `Solaris`. Le référentiel `Solaris` est accessible à partir de `http://localhost/`, port 81, comme spécifié

dans l'instance `solaris` à l'aide du service `pkg/server`. Reportez-vous à “Autorisation des utilisateurs à récupérer des packages par le biais d'une interface HTTP” à la page 31.

1. **Créez une nouvelle instance de serveur de dépôt.**

   Utilisez la sous-commande `add` de la commande`svccfg` afin d'ajouter une nouvelle instance du service `pkg/server` .

   ```
   $ svccfg -s pkg/server add studio
   ```

2. **Vérifiez que vous avez ajouté la nouvelle instance.**

   ```
   $ svcs pkg/server
   STATE  STIME    FMRI
   online 14:54:16 svc:/application/pkg/server:default
   online 14:54:20 svc:/application/pkg/server:studio
   online 14:54:20 svc:/application/pkg/server:solaris
   ```

3. **Définissez le chemin vers le référentiel.**

   Définissez le chemin du répertoire où cette instance du service peut trouver les données de référentiel.

   ```
   $ svccfg -s pkg/server:studio setprop pkg/inst_root=/export/IPSpkgrepos/SolarisStudio
   ```

4. **(Facultatif) Définissez le numéro de port pour la nouvelle instance.**

   ```
   $ svccfg -s pkg/server:studio setprop pkg/port=82
   ```

5. **(Facultatif) Définissez la base de proxy Apache.e.**

   Reportez-vous à “Configuration d'un proxy à préfixe simple” à la page 57 obtenir un exemple de le paramétrage de `pkg/proxy_base`.

6. **Définissez le nom du référentiel et sa description.**

   Vérifiez que le nom et la description sont définis comme cela est indiqué dans “Modificiation des valeurs des propriétés d'un référentiel” à la page 44.

7. **Démarrez le service de référentiel.**

   Redémarrez le service du serveur de dépôt de packages.

   ```
   $ svcadm refresh pkg/server:studio
   $ svcadm enable pkg/server:studio
   ```

8. **Vérifiez que le serveur de référentiel fonctionne.**

   Ouvrez une fenêtre de navigateur sur l'emplacement `http://localhost:82/` .

   Si vous n'avez pas défini le numéro de port, la valeur par défaut est 80. Visualiser votre référentiel dans `http://localhost:80/` ou `http://localhost/` .

Si le numéro de port est également utilisé par une autre instance de `pkg/server`, ajoutez le nom de l'éditeur à l'emplacement pour voir les nouveaux packages. Par exemple, affichez votre référentiel, à l'adresse `http://localhost:81/solarisstudio/`.

9. **Définissez l'origine de l'éditeur.**

   Définissez l'origine de l'éditeur sur l'une des valeurs suivantes :

   - L'emplacement `pkg/inst_root`.

     ```
     $ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/SolarisStudio/ \
     solarisstudio
     ```

   - L'emplacement `pkg/port`.

     ```
     $ pkg set-publisher -G '*' -M '*' -g http://localhost:82/ solarisstudio
     ```

Voir aussi Reportez-vous à “Référentiels multiples sous un domaine” à la page 57 pour plus d'informations sur l'exécution de plusieurs référentiels sous un seul nom de domaine avec des préfixes différents, tels que `http://pkg.example.com/solaris` et `http://pkg.example.com/studio`.

▼ Fourniture de plusieurs référentiels à partir d'un emplacement unique

La plupart de ces étapes de cette procédure doivent être identiques à celles décrite précédemment. Pour plus d'informations, reportez-vous à la procédure précédente.

1. **Créez une nouvelle instance de serveur de dépôt.**

2. **Définissez le chemin vers le référentiel.**

   Chaque instance de `pkg/server` gérée par une instance `pkg/depot` particulière doit disposer d'une valeur `pkg/inst_root` unique.

3. **Vérifiez la propriété `readonly` pour la nouvelle instance.**

   La valeur par défaut de la propriété `pkg/readonly` est `true`. Si cette valeur a été modifiée, réinitialisez la valeur sur `true`.

   ```
   $ svcprop -p pkg/readonly pkg/server:studio
   true
   ```

4. **Définissez la propriété `standalone` pour la nouvelle instance.**

Par défaut, la valeur de la propriété `pkg/standalone` a la valeur `true`. Toutes les instances `pkg/server` dont la propirété `pkg/standalone` est définie sur `false` peut être traitée à partir du même emplacement par une instance de service `pkg/depot`.

```
$ svccfg -s pkg/server:studio
svc:/application/pkg/server:studio> setprop pkg/standalone=false
svc:/application/pkg/server:studio> refresh
svc:/application/pkg/server:studio> select solaris
svc:/application/pkg/server:solaris> setprop pkg/standalone=false
svc:/application/pkg/server:solaris> refresh
svc:/application/pkg/server:solaris> exit
$
```

Assurez-vous que la valeur de la propriété `pkg/inst_root` est unique pour chaque instance de `pkg/server` dont la propriété `pkg/standalone` est définie sur `false`.

5. **(Facultatif) Définissez le numéro de port pour l'instance `pkg/depot`.**

   Par défaut, le numéro de port du service `svc:/application/pkg/depot:default` est 80. Ce numéro de port peut être identique à celui du numéro de port de n'importe quelle instance `pkg/server` gérée par cette instance `pkg/depot`. Pour modifier le numéro de port, définissez la propriété `config/port`par défaut de `pkg/depot:default`.

6. **Redémarrez l'instance `pkg/depot`.**

   ```
   $ svcadm refresh pkg/depot:default
   $ svcadm restart pkg/depot:default
   ```

7. **Vérifiez que le serveur de référentiel fonctionne.**

   Lorsque les utilisateurs ouvrent l'emplacement `http://localhost:80/`, ils voient le référentiel `http://localhost/solaris` indiqué à l'aide de l'éditeur `solaris`, et ils voient le référentiel `http://localhost/studio` indiqué à l'aide de l'éditeur `solarisstudio`.

   Si un seul référentiel fournit les packages pour plusieurs éditeurs, tous les éditeurs sont répertoriés. Par exemple, les utilisateurs risquent de voir le référentiel `http://localhost/solaris` répertorié avec les éditeurs `solaris` et `isvpub`.

8. **Définissez l'origine de l'éditeur.**

   Définissez l'origine de l'éditeur sur l'une des valeurs suivantes :

   - Emplacement `pkg/inst_root` unique.

     ```
     $ pkg set-publisher -G '*' -M '*' -g /export/IPSpkgrepos/SolarisStudio/ \
     solarisstudio
     ```

   - L'emplacement défini par la valeur `config/port` plus le nom d'instance `pkg/server`.

     ```
     $ pkg set-publisher -G '*' -M '*' -g http://localhost:80/studio/ solarisstudio
     ```

Étapes suivantes

Si vous modifiez le contenu d'un référentiel qui est géré par une instance `pkg/depot`, comme présenté dans“Mise à jour du référentiel local” à la page 35 et “Personnalisation du référentiel local” à la page 44, effectuez les deux étapes suivantes :

- Exécution de `pkgrepo refresh` sur le référentiel.
- Exécution de `svcadm restart` sur l'instance `pkg/depot`.

Vous pouvez créer d'autres instances du service `pkg/depot` où chaque instance héberge un ou plusieurs référentiels.

Pour générer une configuration autonome au lieu de configurer les instances de services `pkg/server` et `pkg/depot`, reportez-vous à la page de manuel `pkg.depot-config`(1M) .

♦ ♦ ♦ CHAPITRE 5

Exécution du serveur de dépôt derrière un proxy Web

L'exécution du serveur de dépôt derrière une instance du serveur Web Apache offre les avantages suivants :

- Autorisation d'hébergement de plusieurs référentiels sous un même nom de domaine. Le serveur de dépôt `pkg`(5) vous permet de facilement fournir l'accès à un référentiel dans le réseau local ou sur Internet. Cependant, le serveur de dépôt ne prend pas en charge l'utilisation de plusieurs référentiels sous un nom de domaine ou des préfixes sophistiqués. Pour héberger plusieurs référentiels sous un seul nom de domaine, exécutez le serveur de dépôt derrière un proxy Web.
- Performances et disponibilité L'exécution du serveur de dépôt derrière un proxy Web permet également d'améliorer les performances du serveur en activant l'équilibrage de charge sur plusieurs dépôts et en permettant la mise en cache du contenu.
- Serveur fournissant un référentiel sécurisé permet. Exécutez le serveur de dépôt derrière un SSL (Secure Sockets Layer) instance qui prend en charge protocole Apache activé pour les certificats client.

Configuration Apache du serveur de dépôt

Les exemples de cette section utilisent le serveur Web Apache en tant que logiciel proxy. Pour activer le serveur Web Apache, activez le service `service svc:/network/http:service apache22`. Consultez la page Apache HTTP Server Version 2.2 Documentation pour plus d'informations.

Vous devriez pouvoir appliquer les principes présentés dans ces exemples à n'importe quel logiciel de serveur proxy.

Le système d'exploitation Oracle Solaris 11.2 OS inclut le serveur web Apache dans le package `web/server/apache-22` et un fichier de base `httpd.conf` dans `/etc/apache2/2.2`. En règle générale, vous pouvez utiliser la commande suivante pour localiser le fichier `httpd.conf` :

```
$ pkg search -Hl -o path ':file:path:*httpd.conf'
```

```
etc/apache2/2.2/httpd.conf
etc/apache2/2.2/original/httpd.conf
```

Paramètre de configuration Apache requis

Si vous exécutez le serveur de dépôt de packages derrière une instance du serveur web Apache, incluez le paramètre suivant figurant dans votre fichier `httpd.conf` pour ne pas décoder les barres obliques codées :

```
AllowEncodedSlashes NoDecode
```

Noms des packages peuvent contenir des barres obliques codées par URL car des barres obliques utilisés pour exprimer hiérarchique noms de package. Par exemple, le nom de package `pkg://solaris/developer/build/make` devient `http://pkg.oracle.com/solaris/release/manifest/0/developer%2Fbuild%2Fmake` pour le serveur web. Pour empêcher que ce genre de des barres obliques soit interprété comme des séparateurs de répertoire, demandez à Apache de ne pas décoder les pas des barres obliques `%2F`codées.

L'omission de ce paramètre peut retourner une erreur `404 Not Found` avec un impact très négatif sur la fonctionnalité de recherche.

Paramètres de configuration Apache génériques recommandés

Les paramètres suivants ont une incidence sur les performances et la sécurité.

Réduire la taille des métadonnées câblées.

Les clients HTTP peuvent indiquer au serveur qu'ils acceptent les données compressées dans une demande HTTP. L'activation du filtre Apache DEFLATE peut considérablement réduire la taille sur le réseau de métadonnées telles que les catalogues et manifestes. Les métadonnées telles que les catalogues et manifestes sont souvent compressés à 90 %.

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain
```

Permettre plus de requêtes dans la queue d'attente.

Augmentez la valeur `MaxKeepAliveRequests` pour permettre aux clients d'effectuer un plus grand nombre de demandes en attente sans fermer la connexion.

```
MaxKeepAliveRequests 10000
```

Définir le délai d'attente maximal pour la réponse.

Le délai d'attente du proxy définit la durée pendant laquelle Apache attend la réponse du dépôt d'arrière-plan. Pour la plupart des opérations, une durée de 30 secondes est

satisfaisante. Les recherches livrant un très grand nombre de résultats peuvent être bien plus longues. Vous pouvez souhaiter définir une valeur de délai d'attente plus longue afin de vous adapter à de telles recherches.

```
ProxyTimeout 30
```

Désactiver le proxy de transfert.

Vérifiez que la création de proxy de transfert est désactivée.

```
ProxyRequests Off
```

Configuration de la mise en cache pour le serveur de dépôt

Une configuration minimale est nécessaire pour définir le serveur de dépôt derrière un proxy de mise en cache. A l'exception du fichier d'attributs de catalogue (reportez-vous à “Considérations relatives au cache pour le fichier d'attributs de catalogue” à la page 56) et résultats de recherche du référentiel (reportez-vous à “Considérations relatives au cache pour la recherche” à la page 56), tous les fichiers servis sont uniques et peuvent donc être mis en cache indéfiniment si nécessaire. En outre, toutes les réponses de dépôt contiennent les en-têtes HTTP appropriés pour s'assurer que les fichiers dans le cache ne deviennent pas obsolètes par inadvertance.

Reportez-vous au Caching Guide (Guide de mise en cache) Apache pour plus d'informations sur la configuration d'Apache en tant que proxy de mise en cache.

Utilisez la directive `CacheRoot` pour spécifier le répertoire qui contiendra les fichiers mis en cache. Assurez-vous que le répertoire spécifié est accessible en écriture par le processus Apache. Aucun message d'erreur explicite n'est sorti si Apache ne peut pas écrire dans ce répertoire.

```
CacheRoot /tank/proxycache
```

Apache vous permet d'activer la mise en cache pour des répertoires donnés. Vous souhaiterez peut-être que votre serveur de référentiel mette en cache tout le contenu sur le serveur, comme indiqué dans la directive suivante.

```
CacheEnable disk /
```

Utilisez la directive `CacheMaxFileSize` pour définir la taille maximale des fichiers à mettre en cache. La valeur par défaut de 1 Mo d'Apache risque d'être trop faible pour la plupart des référentiels. La directive suivante définit la taille maximale de fichiers mis en cache à 1 Go.

```
CacheMaxFileSize 1000000000
```

Ajustez la structure de répertoires de la mise en cache sur disque pour des performances optimales avec le système de fichiers sous-jacent. Dans un jeu de données ZFS, la présence de plusieurs niveaux de répertoires affecte davantage les performances que le nombre de

fichiers dans un répertoire. Configurez donc un niveau de répertoire avec un grand nombre de fichiers dans chaque répertoire. Utilisez les directives `CacheDirLevels` et `CacheDirLength` pour contrôler la structure de répertoires. Définissez `CacheDirLevels` sur 1. Définissez `CacheDirLength` sur une valeur permettant d'obtenir un bon équilibre entre le nombre de répertoires et le nombre de fichiers par répertoire. La valeur 2 définie ci-dessous générera 4 096 répertoires. Reportez-vous à la documentation Disk-based Caching (Mise en cache sur disque) Apache pour plus d'informations.

```
CacheDirLevels 1
CacheDirLength 2
```

Considérations relatives au cache pour le fichier d'attributs de catalogue

Le fichier d'attributs de catalogue du référentiel (`catalogue.attrs`) contient l'état actuel du catalogue du référentiel. Ce fichier peut être assez important pour garantir la mise en cache. Toutefois, ce fichier devient obsolète si le catalogue du référentiel d'arrière-plan a changé. Vous pouvez utiliser l'une des deux méthodes suivantes pour résoudre ce problème.

- Ne mettez pas en cache ce fichier. Cette solution fonctionne au mieux si le serveur du référentiel s'exécute dans un environnement avec une large bande passante où le trafic supplémentaire n'est pas un facteur important. Le fichier `httpd.conf` partiel suivant montre comment spécifier de ne pas mettre en cache le fichier `catalog.attrs` :

  ```
  <LocationMatch ".*/catalog.attrs">
          Header set Cache-Control no-cache
  </LocationMatch>
  ```

- Supprimez ce fichier du cache chaque fois que le catalogue du référentiel d'arrière-plan est mis à jour.

Considérations relatives au cache pour la recherche

La recherche d'un référentiel de packages génère des réponses personnalisées en fonction de la demande. Par conséquent, les résultats de recherche ne sont pas bien adaptés à la mise en cache. Le serveur de dépôt définit les en-têtes HTTP appropriés pour s'assurer que les résultats de recherche ne deviennent pas obsolètes dans un cache. Cependant, les économies de bande passante attendues de la mise en cache sont faibles. Le fichier `httpd.conf` partiel suivant montre comment spécifier de ne pas mettre en cache les résultats de recherche.

```
<LocationMatch ".*/search/\d/.*">
        Header set Cache-Control no-cache
</LocationMatch>
```

Configuration d'un proxy à préfixe simple

Cet exemple montre la configuration de base pour un serveur de dépôt à charge non équilibrée. Cet exemple connecte `http://pkg.example.com/myrepo` à `internal.example.com:10000`.

Reportez-vous à la section “Fourniture de plusieurs référentiels par le biais de l'accès à un serveur Web” à la page 47 pour obtenir des instructions sur la définition d'autres propriétés dont vous avez besoin et qui ne sont pas abordées dans cet exemple.

Vous devez configurer le serveur de dépôt avec un paramètre `pkg/proxy_base` qui nomme l'URL où le serveur de dépôt est accessible. Utilisez les commandes suivantes pour définir `pkg/proxy_base` :

```
$ svccfg -s pkg/server add repo
$ svccfg -s pkg/server:repo setprop pkg/proxy_base = astring: http://pkg.example.com/
myrepo
$ svcadm refresh pkg/server:repo
$ svcadm enable pkg/server:repo
```

Le client `pkg`(5) ouvre 20 connexions simultanées vers le serveur de dépôt lorsque qu'il effectue des opérations de réseau. Assurez-vous que le nombre de threads de dépôt correspond à tout moment aux connexions au serveur attendues. Utilisez les commandes suivantes pour définir le nombre de threads par dépôt :

```
$ svccfg -s pkg/server:repo setprop pkg/threads = 200
$ svcadm refresh pkg/server:repo
$ svcadm restart pkg/server:repo
```

Utilisation de `nocanon` pour supprimer la canonicalisation des URL. Ce paramétrage est important dans l'ordre pour que la recherche ne posent pas de problème. En outre, limitez le nombre de connexions d'arrière-plan au nombre de threads fournis par serveur de dépôt. Le fichier `httpd.conf` partiel suivant montre comment mandater un serveur de dépôt :

```
Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ http://internal.example.com:10000/ nocanon max=200
```

Pour plus d'informations sur le proxy SSL du noyau Oracle Solaris et l'utilisation de SSL pour chiffrer et accélérer les communications des serveurs Web, reportez-vous au Chapitre 3, “ Serveurs Web et protocole SSL (Secure Sockets Layer) ” du manuel “ Sécurisation du réseau dans Oracle Solaris 11.2 ”.

Référentiels multiples sous un domaine

La principale raison d'exécuter le serveur de dépôt derrière un proxy est que cela permet d'exécuter facilement plusieurs référentiels sous un seul nom de domaine avec des préfixes

différents. L'exemple de la section “Configuration d'un proxy à préfixe simple” à la page 57 peut facilement être étendu pour prendre en charge plusieurs référentiels.

Dans cet exemple, trois préfixes différents d'un nom de domaine sont connectés à trois référentiels de packages différents :

- `http://pkg.example.com/repo_one` est connecté à `internal.example.com:10000`
- `http://pkg.example.com/repo_two` est connecté à `internal.example.com:20000`
- `http://pkg.example.com/xyz/repo_three` est connecté à `internal.example.com:30000`

Le serveur de dépôt `pkg`(5) est un service géré par SMF. Par conséquent, pour exécuter plusieurs serveurs de dépôt sur le même hôte, il vous suffit de créer une nouvelle instance de service :

```
$ svccfg -s pkg/server add repo1
$ svccfg -s pkg/server:repo1 setprop pkg/property=value
$ ...
```

Comme dans l'exemple précédent, chaque serveur de dépôt s'exécute avec 200 threads.

```
Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ http://internal.example.com:10000/ nocanon max=200

Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000/ nocanon max=200

Redirect /xyz/repo_three http://pkg.example.com/xyz/repo_three/
ProxyPass /xyz/repo_three/ http://internal.example.com:30000/ nocanon max=200
```

Configuration de l'équilibrage de charge

Vous pouvez souhaiter exécuter des serveurs de dépôt derrière un équilibreur de charge Apache. L'équilibrage de charge est l'une, notamment en ce qui afin d'augmenter la disponibilité des votre référentiel. Cette section illustre deux situations dans lesquelles l'équilibrage de charge.

L'équilibrage de charge est appliqué si vos référentiels, les catalogues désignent en dans toutes les doit être exactement le même les référentiels pour éviter les problèmes lorsque le programme d'équilibrage de charge clients à partir d'un seul noeud commutateurs à un autre. Pour vous assurer que les catalogues sont exactement les mêmes, clonez les référentiels qui participent à l'équilibrage de charge comme décrit dans “Gestion de plusieurs référentiels locaux identiques” à la page 39.

Un serveur de référentiel avec équilibrage de charge

Cet exemple connecte `http://pkg.example.com/myrepo` à `internal1.example.com:10000` et `internal2.example.com:10000`.

Configurez le serveur de dépôt avec un paramètre `proxy_base` approprié comme illustré dans la section “Configuration d'un proxy à préfixe simple” à la page 57.

Limitez le nombre de connexions d'arrière-plan au nombre de threads exécutés par chaque dépôt divisé par le nombre de dépôts dans la configuration de l'équilibreur de charge. Dans le cas contraire, Apache ouvre plus de connexions vers un dépôt qu'il n'y en a de disponibles et elles se bloquent, ce qui peut nuire aux performances. Spécifiez le nombre maximal de connexions simultanées vers chaque dépôt par le biais du paramètre `max=`. L'exemple ci-dessous montre deux dépôts, chacun exécutant 200 threads. Reportez-vous à “Configuration d'un proxy à préfixe simple” à la page 57 pour un exemple de la procédure de définition du nombre de threads de dépôt.

```
<Proxy balancer://pkg-example-com-myrepo>
        # depot on internal1
        BalancerMember http://internal1.example.com:10000 retry=5 max=100

        # depot on internal2
        BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /myrepo http://pkg.example.com/myrepo/
ProxyPass /myrepo/ balancer://pkg-example-com-myrepo/ nocanon
```

Un serveur de référentiel avec et un serveur de référentiel sans équilibrage de charge

L'exemple suivant inclut toutes les directives que vous devez ajouter au fichier `httpd.conf` pour un serveur de référentiel hébergeant une configuration de serveur de dépôt à charge équilibrée et non équilibrée.

Dans cet exemple, deux préfixes différents d'un nom de domaine sont connectés à trois référentiels de packages différents :

- `http://pkg.example.com/repo_one` est connecté à `internal1.example.com:10000` et `internal2.example.com:10000`
- `http://pkg.example.com/repo_two` est connecté à `internal1.example.com:20000`

```
AddOutputFilterByType DEFLATE text/html application/javascript text/css text/plain

AllowEncodedSlashes NoDecode
```

```
MaxKeepAliveRequests 10000

ProxyTimeout 30

ProxyRequests Off

<Proxy balancer://pkg-example-com-repo_one>
        # depot on internal1
        BalancerMember http://internal1.example.com:10000 retry=5 max=100

        # depot on internal2
        BalancerMember http://internal2.example.com:10000 retry=5 max=100
</Proxy>

Redirect /repo_one http://pkg.example.com/repo_one/
ProxyPass /repo_one/ balancer://pkg-example-com-repo_one/ nocanon
Redirect /repo_two http://pkg.example.com/repo_two/
ProxyPass /repo_two/ http://internal.example.com:20000/ nocanon max=200
```

Configuration d'un accès HTTPS à un référentiel

Tous les clients peuvent que les packages de téléchargement à partir d'un référentiel de packages sur est HTTP configuré pour traiter. Dans certains cas, vous devez limiter l'accès. L'une des méthodes pour restreindre l'accès au référentiel doit être exécuté SSL du serveur de dépôt derrière une instance qui prend en charge Apache activé pour les certificats client.

Offre les avantages suivants à l'aide de SSL, procédez comme suit :

- Transfert de données de package garantit chiffré entre le client et le serveur
- Vous permet d'accorder l'accès aux référentiels basé sur le client présente sur le certificat au serveur

Référentiel pour configurer un serveur sécurisé, vous devez créer une chaîne de certificat personnalisé, procédez comme suit :

1. Créer une autorité de certification (CA), qui est le chef de la chaîne de certificats.
2. A partir de cette CA établir des certificats pour les clients qui sont autorisés à accéder au référentiel.

Une seule copie du CA est stockée sur le serveur de référentiel. A chaque fois qu'un client présente un certificat au serveur spécifique, ce dernier est vérifié auprès de la CA certificat sur le serveur pour déterminer si l'accès est accordé ou non.

Cette section décrit les étapes suivantes pour créer et configurer la chaîne de certificats pour vérifier la fondation Apache les certificats client serveur frontal, procédez comme suit :

- Créer un keystore

- Créer une autorité de certification pour les certificats client
- Ajouter la configuration SSL au fichier de configuration Apache
- Créer une autorité de certification de serveur autosigné
- Créer un keystore PKCS12

Pour plus d'informations sur les privilèges du serveur web Apache, dans Oracle Solaris, reportez-vous à " Verrouiller les ressources à l'aide de privilèges étendus " du manuel " Sécurisation des utilisateurs et des processus dans Oracle Solaris 11.2 "

Création d'un keystore

Pour gérer les certificats, créez un keystore et les clés. CA le, le keystore CA stocke les clés et les clés et les certificats client.

L'outil utilisé pour la gestion du keystore est `pktool`. Reportez-vous à la page de manuel `pktool`(1) pour plus d'informations.

L'emplacement par défaut du keystore pour la commande `pktool` est `/var/user/`*username*, où *username* est le nom de l'utilisateur de système en cours. Ce peut être problématique keystore par défaut lorsqu'un emplacement est géré par plusieurs utilisateurs keystore. En outre, référentiel de packages IPS peut être affecté qu'à un keystore de gestion pour ne pas créer la confusion chez les certificats dédié. Pour définir un emplacement personnalisé pour le référentiel`pktool` de packages IPS pour le keystore, définissez la variable d'environnement `SOFTTOKEN_DIR`. Réinitialisez la variable `SOFTTOKEN_DIR` comme nécessaire à la gestion de plusieurs keystores.

Utilisez les commandes suivantes pour créer un répertoire pour le keystore. Définition du propriétaire en conséquence, si le groupe et les droits d'accès nécessaires pour gérer le plusieurs keystore les utilisateurs.

```
$ mkdir /path-to-keystore
$ export SOFTTOKEN_DIR=/path-to-keystore
```

L'accès au keystore est protégée par une phrase secrète que vous devez saisir chaque fois que vous appelez la commande `pktool`. Pour la création d'un numéro d'unité logique par défaut keystore de la phrase de passe est `changeme`. N'oubliez pas de modifier la phrase de passe `changeme` pour sécuriser davantage la phrase de passe.

Utilisez la commande suivante pour définir la phrase de passe pour le PIN keystore (:)

```
$ pktool setpin
Enter token passphrase: changeme
Create new passphrase:
Re-enter new passphrase:
Passphrase changed.
$ ls /path-to-keystore
```

```
pkcs11_softtoken
```

Création d'une autorité de certification des certificats client

Le niveau CA certificat correspond au plus haut dans votre chaîne de certificat. Le client CA les certificats sont nécessaires pour la génération et à valider les certificats présentés par les clients d'accès à un référentiel.

Les autorités de certification tiers sont gérés par un tels que de nombreuses sociétés de confiance VeriSign. La gestion de confiance permet aux clients de vérifier l'identité d'un serveur par rapport à l'une de leurs autorités de certification. L'exemple de cette section n'incluent pas vérifier l'identité du serveur de référentiel. Vérification de l'exécution de cet exemple affiche uniquement les certificats client. Par conséquent, cet exemple utilise un certificat auto-signé et pour créer la ne permet pas d'utiliser des CA tiers autorités de certification sécurisées.

Le CA requiert un nom commun (CN, common name). Si vous exécutez un seul référentiel, vous pouvez rétablir les CN par le nom de votre organisation (par exemple, "Oracle Software Delivery"). Si vous disposez de plusieurs référentiels, chaque CA doit avoir son propre référentiel. Si c'est le cas, définissez le CN par un nom identifiant de manière unique le référentiel pour laquelle vous souhaitez créer la CA. Par exemple, si vous avez un référentiel de la version d'un référentiel de prise en charge, et les certificats de la version vers laquelle CA uniquement autorise l'accès aux uniquement le référentiel de version, les certificats à partir de la carte et CA autorise l'accès à la prise en charge le référentiel support.

Pour identifier le certificat dans le keystore, définissez une étiquette descriptive pour le certificat. Il est conseillé de définir l'étiquette du certificat sur *CN*`_ca`, où *CN* est le CN du certificat.

Exécutez les commandes suivantes pour créer le certificat de CA, où *name* est le CN du certificat et *CAlabel* est l'étiquette du certificat :

```
$ pktool gencert label=CAlabel subject="CN=name" serial=0x01
```

Le seront stockées dans la CA keystore. Utilisez la commande suivante pour afficher le contenu de vos keystore, procédez comme suit :

```
$ pktool list
```

Vous devez extraire le certificat CA à partir du keystore lorsque vous configurez Apache comme le décrit la rubrique “Ajout de la configuration SSL au fichier de configuration Apache” à la page 65. Utilisez la commande suivante pour extraire le certificat CA dans un fichier nommé `ca_file.pem` :

```
$ pktool export objtype=cert label=CAlabel outformat=pem \
outfile=ca_file.pem
```

Création de certificats client utilisés pour l'accès au référentiel

Une fois que vous avez généré les CA, vous pouvez générer les certificats client.

Génération d'une demande de signature de certificat

Pour générer un certificat client, générez une demande de signature de certificat (CSR). Le contient toutes les informations CSR que vous avez besoin de communiquer de manière sécurisée pour le serveur.

Si vous voulez uniquement vérifier si le client dispose d'une capacité d'un certificat valide émis par vous-même , vous n'avez pas besoin d'encoder les informations à jour en conséquence. Lorsque le client présente son certificat serveur sur le serveur, le CA valide le certificat de valider la par rapport au certificat client, qui a été généré par vous. Toutefois, SSL requiert un `subject` pour le CSR. Si vous n'avez pas besoin de transmettre une autre information sur le serveur, vous pouvez vous contenter de définir le `subject` au pays dans lequel le certificat a été émis. Par exemple, vous pouvez définir le `subject` à `C=US`.

Il est conseillé de encodent le nom utilisateur dans le certificat du client pour permettre au serveur d'identifier le client. Le nom utilisateur est le nom de l'utilisateur auquel vous allez effectuer au cours d'accès au référentiel. Vous pouvez utiliser la CN à cette fin. Spécifiez une étiquette pour ce CSR afin que vous puissiez rechercher et extraire la clé pour le certificat final comme décrit dans “L'extraction de la clé de certificat” à la page 64.

Utilisez la commande suivante pour générer la CSR, procédez comme suit :

```
$ pktool gencsr subject="C=US,CN=username" label=label format=pem \
outcsr=cert.csr
```

Utilisez la commande OpenSSL suivante pour inspecter le CSR dans le fichier `cert.csr` :

```
$ openssl req -text -in cert.csr
```

Signature du CSR

Le CSR doit être signé par le CA pour créer un certificat. Pour signer le CSR. fournissez les informations suivantes :

- Définissez l'`issuer` du certificat sur la chaîne obtenue que vous aviez utilisée pour le `subject` lors de la création de CA à l'aide de la commande `gencert` comme indiqué à la section “Création d'une autorité de certification des certificats client” à la page 62.

- Définissez le numéro de série hexadécimal. Dans cet exemple, le numéro de série CA a été spécifié comme `0x01`, de sorte que le premier certificat client doit avoir le numéro de série `0x02`. Incrémentez le numéro de série pour chaque nouveau certificat client que vous générez.

  Chaque CA et ses certificat de client descendant possède son propre ensemble de numéros de série. Si vous disposez de CA configurées dans votre keystore, soyez attentif lors de la définition des numéros de série de certificat.
- Définissez `signkey` sur l'étiquette de la CA dans le keystore.
- Définissez `outcert` sur le nom du fichier de certificat. Il est conseillé d'avoir pour nommer le certificat et la clé une fois le référentiel auquel vous voulez accéder.

Utilisez la commande suivante pour signer le CSR, procédez comme suit :

```
$ pktool signcsr signkey=CAlabel csr=cert.csr  \
serial=0x02 outcert=reponame.crt.pem issuer="CN=name"
```

Le certificat est créé dans le fichier *reponame*`.crt.pem`. Utilisez la commande OpenSSL suivants pour examiner le certificat, procédez comme suit :

```
$ openssl x509 -text -in reponame.crt.pem
```

L'extraction de la clé de certificat

La clé de ce certificat d'extraction à partir du keystore. Définissez `label` sur la même valeur d'étiquette que vous avez spécifiée lors de l'exécution de `gencsr` pour générer le CSR à la section “Génération d'une demande de signature de certificat” à la page 63. Utilisez la commande suivante pour exporter la clé du keystore :

```
$ pktool export objtype=key label=label outformat=pem \
outfile=reponame.key.pem
```

Transférez le certificat et la clé aux systèmes client qui doivent accéder au référentiel protégé par SSL.

Autorisation des systèmes client à accéder au référentiel protégé

Pour accéder au référentiel protégé SSL, les systèmes client doivent disposer d'un exemplaire de le certificat et la clé et doit indiquer le certificat et clé à l'intérieur de la configuration de l'éditeur.

Copiez le certificat (*reponame*`.crt.pem`) et la clé (*reponame*`.key.pem`) dans chaque système client. Vous pouvez, par exemple, copiez-les vers le répertoire `/var/pkg/ssl` de chaque client.

Utilisez la commande suivante pour spécifier le certificat dans votre généré la configuration de l'éditeur et la clé, procédez comme suit :

```
$ pkg set-publisher -k reponame.key.pem -c reponame.crt.pem \
-p https://repolocation
```

Notez que SSL HTTPS d'authentification référentiel n'est prise en charge que pour URI. L'authentification SSL n'est pas prise en charge pour les URI de référentiel de fichiers.

Ajout de la configuration SSL au fichier de configuration Apache

Afin d'utiliser l'authentification basée sur le certificat à utiliser pour votre client, configurez d'abord configurer un référentiel de serveur de dépôt de configuration Apache générique tel que décrit dans “Configuration Apache du serveur de dépôt” à la page 53. Ajoutez ensuire les éléments suivants à la fin de votre fichier `httpd.conf` :

```
# Let Apache listen on the standard HTTPS port
Listen 443

# VirtualHost configuration for request on port 443
<VirtualHost 0.0.0.0:443>
        # DNS domain name of the server, needs to match your server certificate
        ServerName pkg-sec.example.com

        # enable SSL
        SSLEngine On

        # Location of the server certificate and key.
        # You either have to get one from a certificate signing authority like
        # VeriSign or create your own CA for testing purposes (see "Creating a
        # Self-Signed CA for Testing Purposes")
        SSLCertificateFile /path/to/server.crt
        SSLCertificateKeyFile /path/to/server.key

        # Intermediate CA certificate file. Required if your server certificate
        # is not signed by a top-level CA directly but an intermediate authority
        # Comment out this section if you are using a test certificate or your
        # server certificate doesn't require it.
        # For more info:
        # http://httpd.apache.org/docs/2.2/mod/mod_ssl.html#sslcertificatechainfile
        SSLCertificateChainFile /path/to/ca_intermediate.pem

        # CA certs for client verification.
        # This is where the CA certificate created in step 3 needs to go.
        # If you have multiple CAs for multiple repos, just concatenate the
        # CA certificate files
        SSLCACertificateFile /path/to/ca_cert.pem
```

```
        # If the client presents a certificate, verify it here. If it doesn't,
        # ignore.
        # This is required to be able to use client-certificate based and
        # anonymous SSL traffic on the same VirtualHost.
        # This statement could also go into the <Location> tags but putting it
        # here avoids re-negotiation which can cause security issues with older
        # servers/clients:
        # http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2009-3555
        SSLVerifyClient optional

        <Location /repo>
                SSLVerifyDepth 1
                # This is the SSL requirement for this location.
                # Requirements can be made based on various information encoded
                # in the certificate. Two variants are the most useful for use
                # with IPS repositories:
                # a) SSLRequire ( %{SSL_CLIENT_I_DN_CN} =~ m/reponame/ )
                #    only allow access if the CN in the client certificate matches
                #    "reponame", useful for different certificates for different
                #    repos
                #
                # b) SSLRequire ( %{SSL_CLIENT_VERIFY} eq "SUCCESS" )
                #    grant access if clients certificate is signed by one of the
                #    CAs specified in SSLCACertificateFile
                SSLRequire ( %{SSL_CLIENT_VERIFY} eq "SUCCESS" )

                # proxy request to depot running at internal.example.com:12345
                ProxyPass http://internal.example.com:12345 nocanon max=500
        </Location>
</VirtualHost>
```

Création d'une autorité de certification de serveur autosigné

Pour des tests, vous pouvez utiliser un certificat de serveur autosigné authority (CA) au lieu d'un CA tiers. La procédure de création d'une CA de serveur autosigné pour Apache est très similaire à celle pour le client pour créer un les certificats CA décrite dans “Création d'une autorité de certification des certificats client” à la page 62.

Exécutez la commande suivante pour créer un serveur CA. Définissez le `subject` au nom DNS du serveur.

```
$ pktool gencert label=apacheCA subject="CN=apachetest" \
serial=0x01
```

Exécutez la commande suivante pour créer un CA CSR d'un serveur, procédez comme suit. Si le serveur est accessible sous plusieurs noms ou que vous vouliez le rendre disponible sous

l'adresse IP directement, utilisez la directive `subjectAltNames` comme décrit dans OpenSSL Autre nom de la documentation dans Nom de sujet alternatif dans la documentation OpenSSL.

```
$ pktool gencsr label=apache subject="CN=pkg-sec.internal.example.com" \
altname="IP=192.168.1.1,DNS=pkg-sec.internal.example.com" \
format=pem outcsr=apache.csr
```

Utilisez la commande suivante pour signer le CSR. Utilisez `server.crt` pour `SSLCertificateFile`.

```
$ pktool signcsr signkey=apacheCA csr=apache.csr serial=0x02 \
outcert=server.crt issuer="CN=apachetest"
```

Utilisez la commande suivante pour extraire la clé. Utilisez`server.key` pour `SSLCertificateKeyFile`.

```
$ pktool export objtype=key label=apache outformat=pem \
outfile=server.key
```

Pour vous assurer que votre client acceptera cette clé de serveur, ajoutez le certificat CA (`apacheCA`) au répertoire des CA acceptées sur le système client et redémarrez le service `ca-certificates` pour créer les liaisons requises pour OpenSSL.

Utilisez la commande suivante : pour extraire le certificat CA

```
$ pktool export label=apacheCA objtype=cert outformat=pem \
outfile=test_server_ca.pem
```

Copiez le certificat CA certificat au répertoire sur le CA ordinateur client, procédez comme suit :

```
$ cp /path-to/test_server_ca.pem /etc/certs/CA/
```

Les certificats CA : redémarrez le service

```
$ svcadm refresh ca-certificates
```

Avant de continuer, vérifiez que votre nouveau certificat a été lié CA. Après l'actualisation, le certificat de service `ca-certificate` reconstruit les liaisons dans le répertoire `/etc/openssl/certs`. Exécutez la commande suivante pour vérifier si votre nouveau certificat a été liée : CA

```
$ ls -l /etc/openssl/certs | grep test_server_ca.pem
lrwxrwxrwx   1 root     root          40 May  1 09:51 e89d96e0.0 -> ../../certs/CA/
test_server_ca.pem
```

La valeur de hachage, `e89d96e0.0`, peut être différent pour vous, étant donné qu'elle est basée sur l'objet de votre certificat.

Création d'un keystore PKCS12 pour accéder à un référentiel sécurisé avec Firefox

Les certificats PEM créés dans"Création de certificats client utilisés pour l'accès au référentiel" à la page 63 permettent d'accès au référentiel sécurisé moyennant `pkg`. Cependant, pour accéder à la BUI (browser user interface), vous devez convertir le certificat et la clé dans un format Firefox pouvez importer. Firefox accepte les keystores PKCS12.

Dans le tableau ci-dessous PKCS12 OpenSSL keystore commande permettant de créer le pour Firefox, procédez comme suit :

```
$ openssl pkcs12 -export -in /path-to/certificate.pem \
-inkey /path-to/key.pem -out name.p12
```

Pour importer le keystore PKCS12 que vous venez de créer, sélectionnez les menus, onglets et boutons Firefox suivants : Edition > Préférences > Avancé > Cryptage > Administration fiscale > Voir > Importer des certificats.

Importer un certificat à la fois.

Exemple présentant des référentiels sécurisés complets

Cet exemple configure, trois référentiels sécurisés appelés `repo1`, `repo2` et `repo3`. Les référentiels `repo1` et `repo2` sont configurés avec des certificats dédiés. Par conséquent, les certificats `repo1` ne fonctionneront pas sur pour `repo2`, et les certificats `repo2` ne fonctionneront pas sur `repo1`. Le référentiel`repo3` est configuré pour accepter les deuxt certificats.

L'exemple part du principe que vous disposez pour votre complet de l'instance Apache certificat du serveur est aussitôt disponible. Si vous ne disposez pas d'un certificat du serveur pour votre instance Apache, reportez-vous aux instructions relatives à la création d'un certificat de test dans"Création d'une autorité de certification de serveur autosigné" à la page 66.

Les trois sont définissables référentiels sont configurés sous `https://pkg-sec.example.com/repo1`, `https://pkg-sec.example.com/repo2` et `https://pkg-sec.example.com/repo3`. Ces référentiels pointenr vers des serveurs de dépôt configurés à `http://internal.example.com` sur les ports 10001, 10002 et 10003respectivement. Assurez-vous que la variable d'environnement `SOFTTOKEN_DIR` est définie correctement selon la description dans"Création d'un keystore" à la page 61.

▼ Configuration de référentiels sécurisés

1. **Créez un certificat CA pour `repo1`.**

```
$ pktool gencert label=repo1_ca subject="CN=repo1" serial=0x01
$ pktool export objtype=cert label=repo1_ca outformat=pem \
outfile=repo1_ca.pem
```

2. **Créez un certificat CA pour `repo2`.**

```
$ pktool gencert label=repo2_ca subject="CN=repo2" serial=0x01
$ pktool export objtype=cert label=repo2_ca outformat=pem \
outfile=repo2_ca.pem
```

3. **Créez un fichier de certificat CA combiné.**

```
$ cat repo1_ca.pem > repo_cas.pem
$ cat repo2_ca.pem >> repo_cas.pem
$ cp repo_cas.pem /path-to-certs
```

4. **Créez une paire certificat / clé client permettant d'autoriser l'utilisateur `myuser` d'accéder au référentiel `repo1`.**

```
$ pktool gencsr subject="C=US,CN=myuser" label=repo1_0001 format=pem \
outcsr=repo1_myuser.csr
$ pktool signcsr signkey=repo1_ca csr=repo1_myuser.csr  \
serial=0x02 outcert=repo1_myuser.crt.pem issuer="CN=repo1"
$ pktool export objtype=key label=repo1_0001 outformat=pem \
outfile=repo1_myuser.key.pem
$ cp repo1_myuser.key.pem /path-to-certs
$ cp repo1_myuser.crt.pem /path-to-certs
```

5. **Créez une paire certificat / clé client permettant d'autoriser l'utilisateur `myuser` d'accéder au référentiel `repo2`.**

```
$ pktool gencsr subject="C=US,CN=myuser" label=repo2_0001 format=pem \
outcsr=repo2_myuser.csr
$ pktool signcsr signkey=repo2_ca csr=repo2_myuser.csr  \
serial=0x02 outcert=repo2_myuser.crt.pem issuer="CN=repo2"
$ pktool export objtype=key label=repo2_0001 outformat=pem \
outfile=repo2_myuser.key.pem
$ cp repo2_myuser.key.pem /path-to-certs
$ cp repo2_myuser.crt.pem /path-to-certs
```

6. **Configurez Apache.**

   Ajoutez les éléments suivants à la fin de configuration SSL de votre fichier `httpd.conf` :

```
# Let Apache listen on the standard HTTPS port
Listen 443

<VirtualHost 0.0.0.0:443>
        # DNS domain name of the server
        ServerName pkg-sec.example.com

        # enable SSL
```

```
        SSLEngine On

        # Location of the server certificate and key.
        # You either have to get one from a certificate signing authority like
        # VeriSign or create your own CA for testing purposes (see "Creating a
        # Self-Signed CA for Testing Purposes")
        SSLCertificateFile /path/to/server.crt
        SSLCertificateKeyFile /path/to/server.key

        # Intermediate CA certificate file. Required if your server certificate
        # is not signed by a top-level CA directly but an intermediate authority.
        # Comment out this section if you don't need one or if you are using a
        # test certificate
        SSLCertificateChainFile /path/to/ca_intermediate.pem

        # CA certs for client verification.
        # This is where the CA certificate created in step 3 needs to go.
        # If you have multiple CAs for multiple repos, just concatenate the
        # CA certificate files
        SSLCACertificateFile /path/to/certs/repo_cas.pem

        # If the client presents a certificate, verify it here. If it doesn't,
        # ignore.
        # This is required to be able to use client-certificate based and
        # anonymous SSL traffic on the same VirtualHost.
        SSLVerifyClient optional

        <Location /repo1>
                SSLVerifyDepth 1
                SSLRequire ( %{SSL_CLIENT_I_DN_CN} =~ m/repo1/ )
                # proxy request to depot running at internal.example.com:10001
                ProxyPass http://internal.example.com:10001 nocanon max=500
        </Location>

        <Location /repo2>
                SSLVerifyDepth 1
                SSLRequire ( %{SSL_CLIENT_I_DN_CN} =~ m/repo2/ )
                # proxy request to depot running at internal.example.com:10002
                ProxyPass http://internal.example.com:10002 nocanon max=500
        </Location>

        <Location /repo3>
                SSLVerifyDepth 1
                SSLRequire ( %{SSL_CLIENT_VERIFY} eq "SUCCESS" )
                # proxy request to depot running at internal.example.com:10003
                ProxyPass http://internal.example.com:10003 nocanon max=500
        </Location>

</VirtualHost>
```

7. Testez l'accès à `repo1`.

```
$ pkg set-publisher -k /path-to-certs/repo1_myuser.key.pem \
-c /path-to-certs/repo1_myuser.crt.pem \
```

```
-p https://pkg-sec.example.com/repo1/
```

8. **Testez l'accès à `repo2`.**

```
$ pkg set-publisher -k /path-to-certs/repo2_myuser.key.pem \
-c /path-to-certs/repo2_myuser.crt.pem \
-p https://pkg-sec.example.com/repo2/
```

9. **Testez l'accès à `repo3`.**

Utilisez le certificat `repo1` pour tester l'accès à `repo3`.

```
$ pkg set-publisher -k /path-to-certs/repo1_myuser.key.pem \
-c /path-to-certs/repo1_myuser.crt.pem \
-p https://pkg-sec.example.com/repo3/
```

Utilisez le certificat `repo2` pour tester l'accès à `repo3`.

```
$ pkg set-publisher -k /path-to-certs/repo2_myuser.key.pem \
-c /path-to-certs/repo2_myuser.crt.pem \
-p https://pkg-sec.example.com/repo3/
```

Indice

H

I

K

M

O

P

R

S

U

V

Z