

Activité 3 SISR – Déploiement d'application avec OCSInventory

Document 1 : le déploiement d'une application

L'installation, la mise à jour et la suppression d'applications font partie du travail quotidien d'un administrateur réseau. Lorsque le parc de machines s'agrandit, il devient très intéressant d'automatiser cette tâche.

**Préalables :**

- Sur le serveur OCS, le déploiement d'application n'est pas activé par défaut ; il faut mettre le paramètre `DOWNLOAD` à "on" (menu "Configuration", onglet "Télédéploiement") ; mettre aussi 15 secondes à `DOWNLOAD_PERIOD_LATENCY` (temps d'attente entre 2 périodes de télé-déploiement).
- Il faut modifier le chemin du répertoire "download" qui est le répertoire par défaut où les agents vont télécharger le paquet ou les fragments de paquet ainsi qu'un fichier XML d'information : dans les variables `DOWNLOAD_URI_FRAG` et `DOWNLOAD_URI_INFO`, changer *localhost* par l'adresse IP du serveur.
- Ajouter au fichier `/etc/apache2/conf-available/ocsinventory-reports.conf` le bloc suivant car dans la nouvelle version d'Apache2, par défaut, il n'est pas possible d'accéder aux dossiers différents de `/usr/share/` et `/var/www/` et cela n'a pas été pris en compte par l'équipe de développement :

```
<Directory /var/lib/ocsinventory-reports/download>
Options Indexes FollowSymLinks
Require all granted
</Directory>
```
- Le déploiement d'applications ne fonctionne qu'avec un serveur WEB sécurisé utilisant le protocole HTTPS basé sur l'authentification SSL, il est donc nécessaire de configurer le serveur HTTPS basé sur l'authentification SSL (voir document 2).

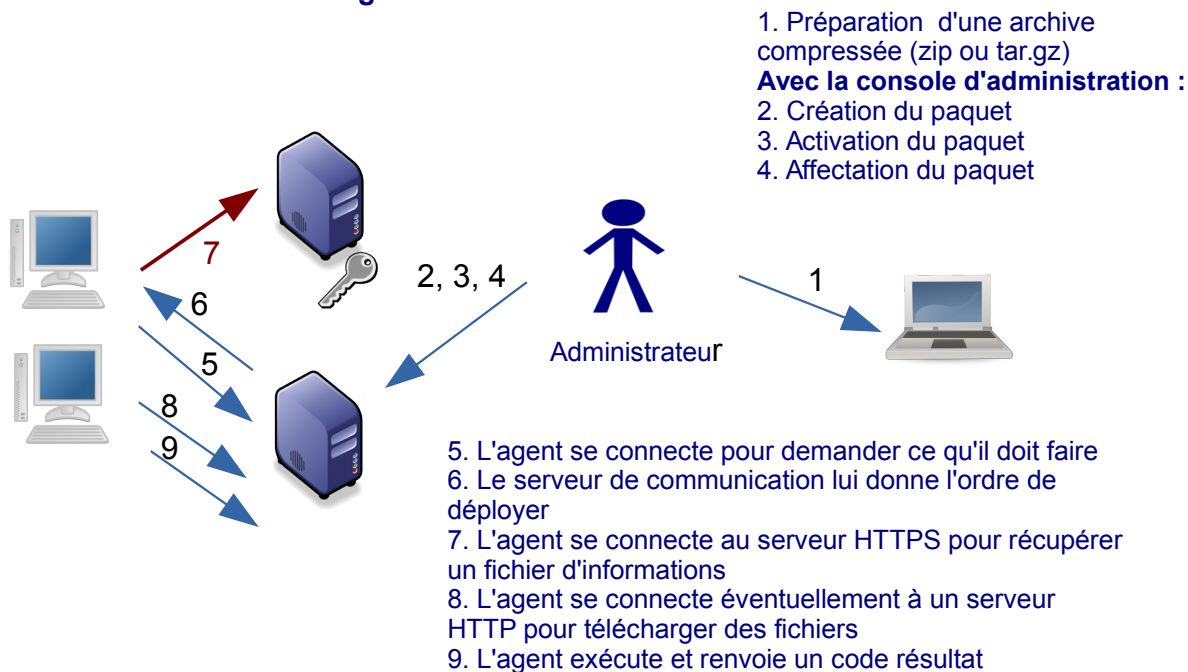
À faire après avoir configuré le serveur HTTPS à partir du document 2

- Modifier la valeur de la variable « `openssl.cafile` » (nouveau paramètre de PHP 5.6) dans le fichier `/etc/php5/apache2/php.ini` (car le fichier recherché par défaut est `/usr/lib/ssl/cert.pem`) :

```
openssl.cafile=/etc/ssl/certs/server.crt
```

Ce paramètre doit être activé en supprimant le « ; » qui le précède.
Pour plus d'informations : <http://php.net/manual/fr/migration56.openssl.php>.
- Copier le certificat généré à partir du document 2 (`server.crt`) dans le répertoire `/etc/ssl/certs/`

Schéma de fonctionnement général :



Le principe de base est le suivant :

- Comme nous l'avons déjà vu, l'agent se connecte au serveur de communication par le protocole HTTP pour lui demander ce qu'il doit faire. En fonction de sa configuration, le serveur peut répondre :
 - d'envoyer un inventaire ;
 - de découvrir le réseau avec le service IpDiscovery ;
 - **de déployer un ou plusieurs paquets ;**
 - de ne rien faire.
- Lorsque l'agent a l'ordre de déployer un paquet, il contacte via le protocole HTTPS le serveur de déploiement afin d'y récupérer un fichier d'informations (IDA : "Instruction Déploiement d'Applications") associé qui est un fichier XML décrivant le paquet et l'action que l'agent devra exécuter. C'est un fichier qui dispose d'un champ d'action important d'où la nécessité de sécuriser et d'authentifier le serveur sur lequel il se trouve.
L'agent devra éventuellement télécharger, via le protocole HTTP, un fichier ou des fragments de fichiers (ce dernier point est optionnel si les instructions ne consistent qu'à exécuter une ou plusieurs commandes).

L'administrateur devra au préalable :

- préparer une archive compressée (en .ZIP pour Windows et en .tar.gz pour Linux) des fichiers nécessaires,
- créer le paquet grâce à la console d'administration,
- activer le paquet,
- affecter le paquet aux machines sur lesquelles le déploiement doit s'effectuer.

Nous partirons d'un exemple simple : nous déploierons sur les postes windows l'utilitaire "putty" (qui ne consiste qu'en un fichier "putty.exe") dans le répertoire "Déploiement" créé préalablement à la racine .

L'outil de déploiement de la barre d'outil est  : , nous allons utiliser les sous-menu "Création" et "Activation".

Création du paquet à déployer

Création d'un nouveau paquet	
Nom:	Utilitaire_putty
Description:	Déploiement de Putty
Système:	WINDOWS ▾
Protocole:	HTTP ▾
Priorité:	5 ▾
Fichier (déployé sur les ordinateurs clients):	Choisissez un fichier putty.zip
Action:	Stocker ▾
Chemin:	C:\Deploiement\
Serveurs de redistribution	
Utilisation sur ce paquet de la redistribution:	NON ▾
Interactions utilisateur	
Prévenir utilisateur:	NON ▾
La fin de l'installation nécessite une intervention utilisateur:	NON ▾
<input type="button" value="Envoyer"/>	

Le nom du paquet à déployer auquel sera associé un identifiant unique dans la base de données.

Le système : Il est possible de déployer des paquets sur Windows ou sur Linux.

Le protocole utilisé pour le transfert des données est HTTP.

La priorité permet de définir quels paquets doivent être installés avant d'autres. Au total 11 niveaux de priorité sont disponibles. Plus le chiffre défini comme priorité est bas, plus la priorité sera forte.

La priorité 0 est donc la plus forte, mais attention, celle-ci doit être utilisée avec précaution car un paquet ayant cette priorité devra obligatoirement se déployer correctement sinon les autres paquets ne seront pas déployés. La priorité 5 (proposée par défaut) convient la plupart du temps.

Le fichier : selon la documentation, tous les paquets doivent être compressés en ZIP pour l'agent Windows et en TAR.GZ pour les ordinateurs Linux. **Cette archive compressée doit donc être préparée préalablement.**

Trois types d'**actions** sont disponibles:

- **Lancer** : pour déployer un fichier ZIP ou TAR.GZ et lancer avec ou sans paramètre un fichier exécutable **incluant** un fichier ZIP ou TAR.GZ.
Le fichier ZIP ou TAR.GZ sera décompressé dans un répertoire temporaire, et la commande associée (le nom du fichier exécutable sans le chemin !) sera lancée dans le répertoire temporaire.
- **Exécuter** : pour déployer un fichier ZIP ou TAR.GZ (optionnellement), et lancer avec ou sans paramètre un fichier exécutable **incluant ou non** un fichier ZIP ou TAR.GZ.
Si l'exécutable n'est pas inclus dans le fichier ZIP ou TAR.GZ, il doit être une partie de logiciel toujours installé dans l'ordinateur client. Typiquement, cela peut être une commande Windows standard tel qu'un appel de l'installateur Windows, commande RPM, DPKG ou TAR.GZ sur Linux.
Le fichier ZIP ou TAR.GZ sera décompressé dans un répertoire temporaire, et la commande associée (le nom du fichier exécutable avec le chemin ou les paramètres si besoin) sera lancée dans le répertoire temporaire.
- **Stocker** : pour déployer un fichier ZIP ou TAR.GZ et enregistrer seulement son contenu dans un enregistrement de l'ordinateur client. Il faut donc donner le chemin de stockage sur l'ordinateur client.

Après un clic sur le bouton de commande "Envoyer", la fenêtre qui doit s'afficher est la suivante :

Création d'un nouveau paquet [Utilitaire_putty]	
Nom du fichier:	putty.zip
Identifiant unique:	1469789473
Digest MD5 / Hexa:	e94250268147cfff8df2de33829718c7
Taille totale:	244 Ko
Taille fragment (1 Ko min):	<input type="text" value="244"/> Ko
Nombre de fragments:	<input type="text" value="1"/>
Temps estimé de déploiement:	<input type="text" value=""/>

Envoyer



Votre paquet a bien été créé dans le répertoire /var/lib/ocsinventory-reports/download/1469789473

Fragmenter un paquet permet de :

- télécharger chaque fragment individuellement et autoriser la reprise sur incident lors d'un gros paquet à télécharger (seul le fragment perdu sera téléchargé à nouveau) ;
- ne pas saturer le réseau.

Mais en contrepartie, s'il y a beaucoup de fragments, le téléchargement peut être long selon les réglages (priorité et paramètres du serveur).

En effet, Le télédéploiement se déroule par période composée de cycles (variable DOWNLOAD_PERIOD_LENGTH à 10 par défaut). A chaque début de cycle, un calcul va être effectué : "numero de cycle" modulo "priorité du paquet" . Si le résultat de ce calcul est égal à 0, alors un fragment du paquet sera téléchargé (donc tous les 5 cycles pour une priorité égale à 5). Une fois le téléchargement de ce fragment terminé, il va y avoir une pause correspondant à DOWNLOAD_FRAG_LATENCY. A chaque fin de cycle, il va aussi y avoir une pause correspondant à DOWNLOAD_CYCLE_LATENCY. Une fois que tous les cycles d'une période se seront déroulés, il y aura encore une pause correspondant à DOWNLOAD_PERIOD_LATENCY avant de passer à la période suivante. Et ainsi de suite... Lorsque tous les fragments seront téléchargés, ils seront rassemblés en un seul fichier qui sera lancé, exécuté ou stocké.

En conclusion, sur un réseau local et si le paquet n'est pas très gros, un seul fragment est préférable à plusieurs.

Et on peut aussi constater que le fichier XML "info" a aussi été créé automatiquement dans ce répertoire :

```
root@serv0csGlpIpi:~# ls -l /var/lib/ocsinventory-reports/download/1469789473/
-rw-r--r-- 1 www-data www-data 250245 jui 29 13:23 1469789473-1
-rw-r--r-- 1 www-data www-data 370 jui 29 13:23 info
```

Un seul fragment

Activation du paquet

Une fois construits les paquets doivent être activés.

Timestamp	Date de création	Nom	Notifié	Succès	Erreur	Archives	Activer	Supprimer	Sélectionner	Stats
1469789473	2016-07-29 12:51:13	Utilitaire_putty							<input type="checkbox"/>	

Activation : MANUELLE ▼

Activation de paquets => Utilitaire_putty (1469789473)

Serveur de fichiers http://192.168.0.150/download/1469789473

Serveur https https://192.168.0.150/download/1469789473



Serveur HTTPS : c'est le serveur où doit être téléchargé le fichier d'instructions XML (info)

Serveur HTTP : c'est le serveur où sera(ont) téléchargé(s) le (ou les) fragment(s)

Ce sont les URL passées en argument à l'agent.



Paquet activé, il peut maintenant être affecté

Fermer


Cliquer sur « Fermer ».

Affectation et déploiement d'un paquet

Une fois le paquet activé, il est possible de le déployer facilement sur un nombre important de postes. Le plus simple est d'effectuer une recherche (dans notre exemple les ordinateurs qui exécutent un système d'exploitation Windows) ou utiliser un groupe dynamique ou statique puis de cliquer sur l'icône "Télédéploier".

Account Info: TAG	Machine(s): Dernier contact	Machine(s): Dernier inventaire	Machine(s): Système	Machine(s): Type agent	Machine(s): Utilisateur	Machine	Supprimer	Sélectionner
MVS	2016-07-29 13:10:10	2016-07-29 13:10:10	Microsoft Windows 7 Professionnel	OCS-NG_WINDOWS_AGENT_v2.0.5.0	apolloie	MVSEVEN64	X	<input type="checkbox"/>

Affectation de masse Pour la sélection ▼

 Action sur le résultat de la sélection

 1 machine(s) affectée(s) avec succès.


Paquets sur les machines **Paquets sur les groupes de redistribution**

Afficher: 20

Restreindre l'affichage: **Filtrer**

Ajouter colonne:

1 Résultat(s) (Télécharger)

Nom du paquet	Priorité	Nom du système	TAILLE (KB)	PACK_LOC	SELECT
Utilitaire_putty	5	WINDOWS	250245	192.168.0.150/download	

Cliquez sur "Select" et les postes clients seront avertis (**notifiés**) qu'un paquet doit être déployé **dès leur prochaine communication avec le serveur (que l'on peut/doit provoquer...)**. Ils téléchargeront alors dans un premier temps le fichier d'instructions XML (info) et ensuite, s'il y a lieu, les différents fragments du paquet.

Le résultat est progressivement visible à partir du sous-menu "Activation" (succès à la fin) :

Timestamp	Date de création	Nom	Notifié	Succès	Erreur	Archives	Activer	Supprimer	Sélectionner	Stats
1469789473	2016-07-29 12:51:13	Utilitaire_putty		1				X	<input type="checkbox"/>	

Des diagrammes sont disponibles pour chaque déploiement de paquet. Ils permettent de disposer d'une image précise du déploiement et de voir quels sont les postes qui ont été notifiés ou pas, ceux dont le déploiement a réussi, ceux qui sont en échec, etc...

Sur le client, les logs vous renseignent sur un éventuel problème :

OCSInventory.log. Vous devriez avoir ces lignes adaptées à votre contexte (à la fin du fichier) :

```
INVENTORY => Writing new inventory state
AGENT => Communication Server ask for Package Download
        DOWNLOAD => Package history file successfully cleaned for duplicate IDs
        DOWNLOAD => Metadata file <info> for package <1469789473> is located at
<https://192.168.0.150/download/1469789473/info>
        COM SERVER => Initializing cURL library for getFile
        COM SERVER => Using cURL without server authentication
        COM SERVER => Disabling cURL proxy support
COM SERVER => Enabling cURL SSL server validation support using CA Bundle <cacert.pem>
        COM SERVER => Sending fileGet request to URL
<https://192.168.0.150/download/1469789473/info>
        COM SERVER => fileGet response received <HTTP Status Code #200>
        COM SERVER => Cleaning cURL library
        DOWNLOAD => Unloading communication provider
        DOWNLOAD => Retrieve info file...OK (pack 1469789473)
DOWNLOAD => Package <1469789473> added to download queue
DOWNLOAD => Download and setup tool successfully started
AGENT => Unloading communication provider
AGENT => Unloading plug-in(s)
AGENT => Execution duration: 00:00:12.
```

Voici un extrait du fichier "download.log".

En observant ce dernier fichier, vous pouvez constater toutes les pauses effectuées (et le temps mis pour le déploiement).

Starting OCS Inventory NG Package Download and Setup Tool on Friday, August 26, 2016 23:15:48.

DOWNLOAD => Running OCS Inventory NG Download Version 2.0.5.0

```
...
DOWNLOAD => Starting new period of 10 cycles
        DOWNLOAD => Flushing package queue
DOWNLOAD => Parsing directory <C:\ProgramData\OCS Inventory NG\Agent\download> for
packages
        DOWNLOAD => Verifying package <C:\ProgramData\OCS Inventory
NG\Agent\download\1469814614>
DOWNLOAD => Package <1469814614> verified and added to process queue
        DOWNLOAD => Processing packages for cycle 1 on Friday, August 26, 2016 23:15:48
        DOWNLOAD => Pausing for cycle latency (20 seconds)
...
        DOWNLOAD => Processing packages for cycle 5 on Friday, August 26, 2016 23:17:08
DOWNLOAD => Downloading package fragment <1469814614-1>
...
        COM SERVER => Enabling cURL SSL server validation support using CA Bundle
<cacert.pem>
        COM SERVER => Sending fileGet request to URL
<HTTP://192.168.0.150/download/1469814614/1469814614-1>
        COM SERVER => fileGet response received <HTTP Status Code #200>
        COM SERVER => Cleaning cURL library
...
        DOWNLOAD => Pausing for cycle latency (20 seconds)
        DOWNLOAD => Processing packages for cycle 10 on Friday, August 26, 2016 23:18:58
DOWNLOAD => Building package <1469814614>
...
        PACKAGE => Verifying ZIP signature for package <1469814614>
        DOWNLOAD => Package <1469814614> built successfully
...
DOWNLOAD => Execution duration: 00:03:51.
```