

Zéphir - Tâche #26615

Scénario # 26518 (Terminé (Sprint)): La sauvegarde restauration de zephir ne fonctionne pas de 2.6.2 vers 2.7.0

procédures utilisées pour restaurer un zephir

28/01/2019 15:35 - Thierry Bertrand

Statut:	Fermé	Début:	05/12/2018
Priorité:	Normal	Echéance:	
Assigné à:	Thierry Bertrand	% réalisé:	100%
Version cible:	sprint 2019 5-7 Equipe MENSUR	Temps estimé:	0.00 heure
		Temps passé:	0.00 heure
Description			

Historique

#1 - 28/01/2019 16:02 - Thierry Bertrand

La procédure pour restaurer un zephir se fait en plusieurs étapes :

- sauvegarde.sh sur le zephir d'origine puis externalisation : on obtient sauvegarde2.6.2.tar.gz
- installation fresh install d'un zephir 2.7.0
- reimport de la sauvegarde sous /var/lib/zephir_backups/
- apt-eole install supervision-psin eole-zephir-medde
- import du certificat racine ayant signé le ldap sous /etc/certs/

A partir de ce moment là, il est nécessaire d'avoir un zephir d'instancié pour que la procédure de restauration fonctionne. Il existe alors 2 méthodes :

A partir d'une conf minimale et d'une base locale

- via gen_config, on fait une conf minimale avec une base de conf locale
- instanciation

On obtient à l'issue un zephir opérationnel mais pas dans l'environnement cible

- restauration.sh
Là la procédure de restauration s'initialise mais plante en disant que le fichier ldap.ldif est vide ou absent. C'est normal puisque la sauvegarde est issue d'une base externe donc son ldap.ldif est vide. Du coup, il va falloir tricher...
- sauvegarde.sh sur le zephir fraîchement instancié, on obtient sauvegarde2.7.0.tar.gz
- sous /var/lib/zephir_backups, tar xzf sauvegarde2.7.0.tar.gz
- on récupère alors le ldap.ldif de la base locale
- sous /var/lib/zephir_backups, tar xzf sauvegarde2.6.2.tar.gz
- on injecte le ldap.ldif précédent
- sous /var/lib/zephir_backups, tar czf sauvegarde2.6.2.tar.gz
- restauration.sh qui va alors jusqu'au bout
- instance (en faisant attention à l'ip)
Si tout se passe bien, on a un zephir restauré mais en 2.7.0

A partir d'une conf ldap externe

L'idée est de repartir du config.eol du zephir 2.6.2

- sous /var/lib/zephir_backups, tar xzf sauvegarde2.6.2.tar.gz
- tar xf etc_eole
- cp /var/lib/zephir/backups/sauvegarde2.6.2/config.eol /etc/eole/
- instance
Là les ennuis commencent :
- vi /usr/share/eole/posttemplate/10-conf-zephir
mise en commentaire de la partie d'authentification ldap

```
# test de bind ldap
#DN=$(ldapsearch -LLL -x -h $adresse_ip_ldap -b "$ldap_base_dn" "uid=$admin_zephir" dn | p
erl -00lne '($dn = $_) =~ s/(?:\^dn:\s+)|(?:\r?\n\s)//g; print $dn') &>/dev/null
#ldapsearch -x -h $adresse_ip_ldap -b "$ldap_base_dn" -D "${DN}" -w "${pass_zeph}" "uid=$a
```

```
admin_zephir" &> /dev/null
    #if [ $? -eq 0 ]
    #then
        pass_ok=1
    #else
    #     EchoOrange "Erreur d'authentification LDAP"
    #fi
```

On garde la partie `pass_ok=1` pour enchaîner les traitements suivants. En fait ici, la requête d'authentification ldap échoue et fait planter l'instance sinon.

Une requête formulée sous cette forme est pourtant correcte :

```
ldapsearch -x -Z -d8 -H ldap://ldapap.m2.e2.rie.gouv.fr:389 -D "uid=admin_zephir,ou=admin,ou=ressources,dc=
=equipement,dc=gouv,dc=fr" -W -b "dc=equipement,dc=gouv,dc=fr" uid=admin_zephir
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <dc=equipement,dc=gouv,dc=fr> with scope subtree
# filter: uid=admin_zephir
# requesting: ALL
#
# admin_zephir, admin, ressources, equipement.gouv.fr
dn: uid=admin_zephir,ou=admin,ou=ressources,dc=equipement,dc=gouv,dc=fr
objectClass: top
objectClass: person
objectClass: uidObject
cn: admin_zephir
uid: admin_zephir
sn: admin_zephir
userPassword:: *****
=
# search result
search: 3
result: 0 Success
# numResponses: 2
# numEntries: 1
```

Cette modification, faite l'instance va jusqu'au bout même si on se fait engueuler sur les certificats...

- restauration de la base 2.6.2
 - instance
- Là si tout se passe bien, on obtient un zephir fonctionnel.

Dans les deux cas, il a fallu modifier le code de `/usr/lib/python2.7/dist-packages/zephir/backend/modules_rpc.py` et remplacer la fonction `startup` par :

```
def startup(self):
    old_obs = log.theLogPublisher.observers
    if old_obs:
        self.old_obs = log.theLogPublisher.observers[0]
    else:
        self.old_obs = None
    new_obs = syslog.SyslogObserver('zephir_backend', options=syslog.DEFAULT_OPTIONS, facility=syslog.DEFA
ULT_FACILITY)
    log.addObserver(new_obs.emit)
    if self.old_obs:
        log.removeObserver(self.old_obs)
    self.xmlrpc_update_modules()
```

Un paquet a été fait par gnunux avec cette correction.

#2 - 14/02/2019 14:46 - Thierry Bertrand

- Statut changé de Nouveau à Fermé
- Restant à faire (heures) mis à 0.0

Au final, les installations se sont faites de la manière suivante :
clonage d'une vm zephir 2.7.0 opérationnel
rapatriement de la sauvegarde d'un autre zephir

#3 - 19/02/2019 16:38 - Joël Cuissinat

- % réalisé changé de 0 à 100
- Temps estimé mis à 0.00 h