

Migration vers EOLE 2.9

EOLE 2.9



Version : révision : Février 2023

Date : création : Décembre 2022

Editeur : Pôle national de compétences Logiciels Libres

Auteur(s) : Équipe EOLE

Copyright : Documentation sous licence Creative Commons by-sa - EOLE <https://pctl.ac-dijon.fr/pctl/>)

Licence : Cette documentation, rédigée par le Pôle national de compétences Logiciels Libres, est mise à disposition selon les termes de la licence :
Creative Commons Attribution - Partage dans les Mêmes Conditions 3.0 France (CC BY-SA 3.0 FR) : <http://creativecommons.org/licenses/by-sa/3.0/fr/>.

Vous êtes libres :

- de **reproduire, distribuer et communiquer** cette création au public ;
- de **modifier** cette création.

Selon les conditions suivantes :

- **Attribution** : vous devez citer le nom de l'auteur original de la manière indiquée par l'auteur de l'œuvre ou le titulaire des droits qui vous confère cette autorisation (mais pas d'une manière qui suggérerait qu'ils vous soutiennent ou approuvent votre utilisation de l'œuvre) ;
- **Partage des Conditions Initiales à l'Identique** : si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

À chaque réutilisation ou distribution de cette création, vous devez faire apparaître clairement au public les conditions contractuelles de sa mise à disposition. La meilleure manière de les indiquer est un lien vers cette page web.

Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits sur cette œuvre.

Rien dans ce contrat ne diminue ou ne restreint le droit moral de l'auteur ou des auteurs.

Cette documentation est basée sur une réalisation du Pôle national de compétences Logiciels Libres. Les documents d'origines sont disponibles sur le site.

EOLE est un projet libre (Licence GPL).

Il est développé par le Pôle national de compétences Logiciels Libres du ministère de l'Éducation nationale, rattaché à la Direction des Systèmes d'Information de l'académie de Dijon (DSI).

Pour toute information concernant ce projet vous pouvez nous joindre :

- Par courrier électronique : eole@ac-dijon.fr
- Par FAX : 03-80-44-88-10
- Par courrier : EOLE-DSI - 2G, rue du Général Delaborde - 21000 DIJON
- Le site du Pôle national de compétences Logiciels Libres : <https://pctl.ac-dijon.fr/pctl/>

Table des matières

Chapitre 1 - Les procédures de migrations vers 2.9	4
Chapitre 2 - Passage d'une version RC à une version stable	5
Chapitre 3 - Migration de 2.7 vers 2.9	6
1. Migration sans réinstallation vers EOLE 2.9	7
2. Migration avec réinstallation vers EOLE 2.9	10
Chapitre 4 - Migration de 2.8 vers 2.9	14
1. Migration sans réinstallation vers EOLE 2.9	14
2. Migration avec réinstallation vers EOLE 2.9	17
Chapitre 5 - Migration de 2.9.n vers 2.9.n+x : Maj-Release	20
Chapitre 6 - Migration vers le module Sphinx 2.9	22
1. Migration sans réinstallation vers EOLE 2.9	22
2. Sauvegarde et restauration	24
3. Migration de 2.9.n vers 2.9.n+x : Maj-Release	26
Chapitre 7 - Migration vers le module Zéphir 2.9	28
Chapitre 8 - Gestion de la migration avec un serveur Zéphir	29
1. Généralités sur la migration	29
2. Préparation de la migration depuis l'application Zéphir	31
3. Migration après réinstallation d'un serveur	38
4. Migration par mise à jour avec les procédure Upgrade-Auto / Maj-Release	39
Glossaire	40

Chapitre 1

Les procédures de migrations vers 2.9



Il existe plusieurs procédures pour migrer vers EOLE 2.9 mais leur disponibilité dépend du module à migrer et de sa version actuelle.

Version du module	Eolebase, Amon, Scribe	Sphynx	Zéphir	AmonEcole
2.7.2 → 2.8.1 → 2.9	Upgrade-Auto	Upgrade-Auto	Upgrade-Auto	---
2.7.n → 2.9	migration29.sh	sauvegarde.sh restauration.sh	sauvegarde.sh restauration.sh	—
2.8.1 → 2.9	Upgrade-Auto	Upgrade-Auto	Upgrade-Auto	Upgrade-Auto
2.8.n → 2.9	migration29.sh	sauvegarde.sh restauration.sh	sauvegarde.sh restauration.sh	migration29.sh
2.9.n → 2.9.n+x	Maj-Release			

La commande `Upgrade-Auto` est disponible uniquement sur la dernière version mineure (ex : 2.7.2, 2.8.1).

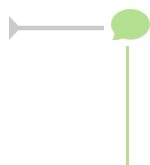
La migration à l'aide d'un serveur Zéphir fait l'objet d'une procédure particulière.

Chapitre 2

Passage d'une version RC à une version stable

Avant d'être publiée en version RC, la distribution Linux EOLE subit de nombreux tests. Aussi, elle ne contient plus aucun changement qui ne peuvent être résolus par mise à jour.

Il est donc possible d'installer une version EOLE RC, de la tester, de l'utiliser et de la mettre à jour pour être au même niveau de mise à jour que la version stable une fois que cette dernière version est publiée. La mise à jour s'effectue avec la commande `Maj-Auto`.



Les versions RC portent un numéro, il signifie uniquement qu'une image ISO a été re-générée, un nombre conséquent de paquets ont été recompilés et cela évite une trop grosse mise à jour.

Chapitre 3

Migration de 2.7 vers 2.9

Procédure avec `Upgrade-Auto` successifs

Le serveur doit être en version EOLE **2.7.2** à jour.

Dans le cas contraire vous devez mettre à niveau votre serveur EOLE 2.7 à l'aide des commandes `Maj-Release` et/ou `Maj-Auto`.

Vous pouvez mettre à jour vers la version 2.9 sans avoir à réinstaller le serveur avec un enchaînement d'exécutions de la commande `Upgrade-Auto` qui permettent de passer successivement :

1. de la version mineure 2.7.2 à la version mineure 2.8.1 ;
2. de la version mineure 2.8.1 à une version 2.9.

En fonction du module et de la quantité de données, cette procédure peut avoir un temps d'exécution beaucoup plus long que la migration avec le script `migration29.sh`.

Quelle est la version mineure de mon serveur ?

Vous pouvez connaître la version mineure de votre serveur avec la commande : `CreoleGet eole_release`.

Procédure avec réinstallation du serveur

Vous pouvez préférer migrer un serveur avec réinstallation pour plusieurs raisons :

- Vous changez le matériel ;
- Vous avez des partition(s) quasiment pleine(s) ;
- Vous avez peu de données ;
- Votre serveur est à une version mineure inférieure à 2.7.2 et vous ne souhaitez pas mettre à niveau vers 2.7.2 avant `Upgrade-Auto`.

Ainsi, pour passer d'une version 2.7 à 2.9, vous pouvez réinstaller votre serveur en utilisant le script `migration29.sh`.

En fonction du module et de la quantité de données, cette procédure peut avoir un temps d'exécution beaucoup plus court qu'avec `Upgrade-Auto` mais nécessite un accès physique au serveur.

Pour migrer les modules Sphinx et Zéphir, il faut utiliser les scripts `sauvegarde.sh` et `restauration.sh`.

1. Migration sans réinstallation vers EOLE 2.9

Procédure avec `Upgrade-Auto` successifs

Le serveur doit être en version EOLE 2.7.2 à jour.

Dans le cas contraire vous devez mettre à niveau votre serveur EOLE 2.7 à l'aide des commandes `Maj-Release` et/ou `Maj-Auto`.

Vous pouvez mettre à jour vers la version 2.9 sans avoir à réinstaller le serveur avec un enchaînement d'exécutions de la commande `Upgrade-Auto` qui permettent de passer successivement :

1. de la version mineure 2.7.2 à la version mineure 2.8.1 ;
2. de la version mineure 2.8.1 à une version 2.9.

En fonction du module et de la quantité de données, cette procédure peut avoir un temps d'exécution beaucoup plus long que la migration avec le script `migration29.sh`.

Quelle est la version mineure de mon serveur ?

Vous pouvez connaître la version mineure de votre serveur avec la commande : `CreoleGet eole_release`.

Passage de la dernière version mineure 2.n vers la dernière version mineure de 2.n+1

DKMS

La procédure de migration refusera de s'exécuter si elle détecte des pilotes compilés (DKMS [p.40]).

Les DKMS sont en effet susceptibles de faire échouer la procédure : impossibilité de démarrer sur le nouveau noyau, fichier présent dans le paquet DKMS fourni par un autre paquet en standard...

Pour des structures avec un faible débit réseau il est possible de limiter la taille du téléchargement en utilisant une image ISO stockée sur une clef USB, un cédérom ou un dépôt local. Dans ce cas, seuls les paquets plus récents que ceux présents sur l'image ISO seront téléchargés.

- `Upgrade-Auto --cdrom` permet de copier le contenu du nouveau CD d'installation EOLE et évite le téléchargement de l'image ISO et des paquets déjà présents sur le CD.
- `Upgrade-Auto --download` permet de procéder uniquement au téléchargement de l'image ISO de la version cible. La migration n'est effectuée qu'après un nouvel `Upgrade-Auto`.
- `Upgrade-Auto --iso <chemin de l'image ISO>` permet de copier le contenu d'une image ISO d'installation EOLE locale. Cela évite son téléchargement et celui des paquets déjà présents sur le CD.
- `Upgrade-Auto --iso <URL de l'iso>`, permet de télécharger l'ISO et les fichiers

connexes (le MD5, la signature, ...) depuis une adresse ou un dépôt particulier.

- Ajouter l'option `--download` à la commande `Upgrade-Auto --cdrom` permet de copier le contenu du nouveau CD d'installation EOLE. La migration n'est effectuée qu'après un nouvel `Upgrade-Auto`.
- Ajouter l'option `--download` à la commande `Upgrade-Auto --iso <chemin de l'image ISO>` permet de procéder uniquement à la copie de l'image ISO. La migration n'est effectuée qu'après un nouvel `Upgrade-Auto`.
- L'option `--limit-rate <bande passante>` permet de personnaliser la limite de la bande passante à utiliser pour le téléchargement. Sa valeur est par défaut fixée à `120k` (120 kilooctets). Cette option est passée directement à la commande `wget`, la valeur `0` désactive la limitation.

Exemples d'utilisation

```
# Upgrade-Auto --limit-rate 0
# Upgrade-Auto --limit-rate 120k
# Upgrade-Auto --download --limit-rate 10M
# Upgrade-Auto --iso /mnt/images/eole-2.7.2-alternate-amd64.iso
# Upgrade-Auto --iso http://mon.miroir.local/pub/iso/EOLE-2.8/2.8.1/eole-2.8.1-alternate
```

Consulter le manuel de la commande pour voir toutes les options :

```
# man Upgrade-Auto
```

Utiliser un fichier meta-release-lts alternatif

Le fichier `meta-release-lts` contient les adresses de tous les dépôts Ubuntu.

Dans un environnement sans accès direct à internet, on peut définir, via ce fichier, l'emplacement interne des dépôts à utiliser par `Upgrade-Auto`.

Dans l'interface de configuration du module, en mode expert, aller dans l'onglet **Dépôt tiers**.

Chemin de téléchargement des informations de changement de version EOLE

La variable `upgrade_auto_meta_release_repo` permet de définir où est téléchargé le fichier `meta-release-lts`.

Upgrade-Auto pour les applications tierces

Vous pouvez inclure l'upgrade d'applications annexes.

Pour cela, il vous faudra créer un fichier `.txt` à l'emplacement suivant `/usr/share/eole/upgrade/source.list.d/`.

Puis indiquer à l'intérieur la version cible (donc la version vers laquelle vous voulez faire l'upgrade), le dépôt actuel, et le nouveau dépôt (celui qui permet d'avoir la version souhaitée) séparés par des ";" :

Exemple

Créer le fichier `/usr/share/eole/upgrade/source.list.d/mtes.txt` avec les lignes suivantes :


```
2.7.2;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu xenial  
main;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu bionic  
main
```

```
2.7.2;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu xenial  
main;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu bionic  
main
```

Puis lancer la commande `Upgrade-Auto`. Toutes les applications installées depuis le dépôt source seront mises à jour vers leur version du dépôt de destination.

Explication du contenu : 2.7.2 est la version vers laquelle nous souhaitons migrer, le premier ppa inscrit est celui qui est actuellement sur le serveur, le second est le ppa souhaité pour la version 2.7.2.

Mise à disposition

Le script `Upgrade-Auto` est disponible directement sur le serveur EOLE après mise à jour de celui-ci.

Fonctionnement général

Le script `Upgrade-Auto` permet d'effectuer la migration de façon automatique.

Les étapes sont :

1. choix de la version EOLE de destination ;
2. mise à jour des fichiers `sources.list` ;
3. téléchargement des paquets ;
4. installation des paquets ;
5. mise à jour automatique des fichiers de configuration (`*.eol`) ;
6. mise à jour des paramètres du serveur sur le serveur Zéphir si le serveur y est enregistré.

Procédure détaillée

Pré-requis

Par précaution, il est recommandé de faire une sauvegarde du serveur avant de lancer le script de migration.

La mise à jour du système n'est pas automatisée car elle est susceptible de poser des questions à l'utilisateur en cours d'exécution ;

DKMS

La procédure de migration refusera de s'exécuter si elle détecte des pilotes compilés (DKMS [p.40]).

Les DKMS sont en effet susceptibles de faire échouer la procédure : impossibilité de démarrer sur le nouveau noyau, fichier présent dans le paquet DKMS fourni par un autre paquet en standard...



L'utilisation du script de migration est critique, il n'y a pas de retour en arrière possible une fois les paquets installés.

Réalisation de la migration

1. Mettre à jour le serveur EOLE dans la dernière version mineure :

```
# Maj-Auto
```

2. Reconfigurer et re-démarrer le serveur (si nécessaire) :

```
# reconfigure
```

```
# reboot
```

3. Lancer le script de migration :

```
# Upgrade-Auto
```

4. Si le module utilise les conteneurs, comme c'est le cas pour AmonEcole, il faut recréer les conteneurs :

```
# gen_conteneurs
```

5. Convertir le fichier `config.eol` du module sauvegardé

- Si le serveur est enregistré sur un module Zéphir, il est possible de préparer la configuration et de la récupérer avec la commande `enregistrement_zephir` :

```
# enregistrement_zephir
```

- Sinon, exécuter l'interface de configuration du module, vérifier les valeurs et enregistrer.

6. Instancier et re-démarrer le serveur (si nécessaire) :

```
# instance
```

```
# reboot
```



Screen est une console virtuelle qui permet de détacher un terminal du processus appelant. Il permet notamment de récupérer la console virtuelle en cas de coupure du tunnel SSH.

Pour utiliser Screen :

```
# screen
```

Pour ouvrir une nouvelle console virtuelle :

```
Ctrl + a puis c
```

Pour passer d'une console à l'autre :

```
Ctrl + a + puis barre espace
```

Pour mettre Screen en tâche de fond :

```
# screen -D
```

Pour récupérer une session Screen :

```
# screen -x
```

2. Migration avec réinstallation vers EOLE 2.9

`migration29.sh` est le script de migration qui permet de passer des versions EOLE 2.7.2 ou 2.8.1 à la version 2.9 avec réinstallation du serveur.

Mise à disposition

Le script `migration29.sh` est mis à disposition sur le site FTP du projet EOLE.

Il est fonctionnel pour les modules 2.7.2 et 2.8.1 : Eolebase, Amon et Scribe ainsi que pour le module AmonEcole 2.8.1.

Fonctionnement général

L'utilisation d'un support externe est recommandé.

Le script permet de réaliser une sauvegarde puis une restauration spécifiques.

Des traitements sont effectués sur les données au cours des 2 opérations.

Procédure détaillée

Prérequis

Pour des raisons pratiques, la migration nécessite l'utilisation d'un support externe (partage distant ou disque USB) disposant d'un espace libre suffisant et d'un système de fichiers supportant les liens symboliques (ext3 conseillé).

Il est très fortement conseillé :

- de ne pas utiliser le même serveur que votre module 2.7/2.8 pour accueillir votre module EOLE 2.9 (ou tout du moins pas le même disque dur)
- de posséder une sauvegarde récente du serveur en version 2.7/2.8.

Le serveur 2.7/2.8 doit avoir été mis à jour, reconfiguré et redémarré si nécessaire.

Sauvegarde spéciale du module en version 2.7.2 ou 2.8.1

1. Télécharger le script de migration :

```
# wget ftp://eole.ac-dijon.fr/pub/Outils/migration/migration29.sh
```

2. Lancer la sauvegarde spécifique :

```
# sh migration29.sh
```

3. Éteindre ou débrancher le serveur du réseau

Restauration spéciale du module en version 2.9

1. Installer le module à partir de la dernière image EOLE 2.9

2. Mettre à jour le module :

```
# Maj-Auto
```

3. Si le module utilise les conteneurs, comme c'est le cas pour AmonEcole, il faut recréer les conteneurs :

```
# gen_conteneurs
```

4. Convertir le fichier `config.eol` du module sauvegardé

- Si le serveur est enregistré sur un module Zéphir, il est possible de préparer la configuration et de la récupérer avec la commande `enregistrement_zephir` :

```
# enregistrement_zephir
```

- Sinon, placer l'ancien fichier de configuration avec l'extension `.eol` dans le répertoire `/root` (pour faciliter sa sélection depuis le snap firefox, par exemple), exécuter la commande `gen_config` pour lancer l'interface de configuration du module, utiliser la fonctionnalité : `Fichier -> Importer une configuration`, vérifier les valeurs et enregistrer.

5. Instancier le serveur à l'aide du fichier converti :

```
# instance
```

6. Télécharger le script de migration :

```
# wget ftp://eole.ac-dijon.fr/pub/Outils/migration/migration29.sh
```

7. Lancer la restauration spécifique :

```
# sh migration29.sh
```

8. Reconfigurer le serveur :

```
# reconfigure
```



Auparavant, il était nécessaire de faire attention au répertoire de copie de l'ancien fichier de configuration pour l'étape de conversion. Les restrictions de droits pour l'application de configuration du module ne s'appliquent plus à partir de la version 2.9.0 du module.

Éléments migrés

Éléments communs

- les certificats SSL ;
- les personnalisations SSO ;
- l'adresse du serveur de test de bande passante ;
- les rôles personnalisés via l'EAD ;
- la désactivation de la mise à jour hebdomadaire.

Éléments spécifiques à Amon/AmonEcole

- Personnalisations e2guardian :
 - bases filtres activées ;
 - sites, extensions et types mimes interdits ;
 - sites autorisés ;
 - mode de filtrage ;
- Personnalisations Squid :
 - domaines non mis en cache ;
 - domaines non soumis à l'authentification ;
- Personnalisations du routage :
 - groupes de machine ;
 - destinations interdites ;
 - sources interdites ;
 - règles optionnelles ;
 - horaires du pare-feu.

Éléments spécifiques à Scribe

- l'annuaire et le `.reader` ;
- l'annuaire Active Directory (selon version) ;

- les données liées à Samba et à CUPS ;
- les données (`/home`) ;
- les données du serveur Bcdi Web ;
- les boîtes aux lettres locales ;
- les listes de diffusion ;
- les quotas utilisateur ;
- les ACL sur les données ;
- certaines bases MySQL (sympa) ;
- la configuration Bareos ;
- la configuration WPKG ;
- les modèles de partage ;
- les adresses réservées dans le DHCP.

⚠ Applications web

La restauration des données liées aux applications web Envoie fera l'objet de scripts dédiés.

Suivi du script

Le script `migration29.sh` est versionné dans le dépôt Git Creole (branche *migration*) :

<http://dev-eole.ac-dijon.fr/gitweb/gitweb.cgi?p=creole.git;a=tree;h=refs/heads/migration;hb=refs/heads/mi>

Chapitre 4

Migration de 2.8 vers 2.9

La procédure avec Upgrade-Auto

Le serveur doit être en version mineure 2.8.1 à jour, dans le cas contraire vous pouvez mettre à niveau votre serveur avec la commande `Maj-Release`.

Vous pouvez mettre à jour vers la version 2.9 sans avoir à réinstaller le serveur avec la commande `Upgrade-Auto`.

En fonction du module et de la quantité de données, cette procédure peut avoir un temps d'exécution beaucoup plus long que la migration avec le script `migration29.sh`.

⚡ Quelle est la version mineure de mon serveur ?

Vous pouvez connaître la version mineure de votre serveur avec la commande : `CreoleGet eole_release`.

La réinstallation du serveur est possible

Vous pouvez préférer migrer un serveur avec réinstallation pour plusieurs raisons :

- Vous changez le matériel ou vous avez des partition(s) quasiment pleine(s) ;
- Vous avez peu de données et un accès physique à la machine à migrer (ex : *Amon*) ;
- Votre serveur est à une version 2.8 inférieure à la version 2.8.1 et vous ne souhaitez pas faire de `Maj-Release` avant `Upgrade-Auto` ;

En fonction du module et de la quantité de données, cette procédure peut avoir un temps d'exécution beaucoup plus court qu'avec `Upgrade-Auto` mais nécessite un accès physique au serveur.



Pour migrer les modules Sphinx et Zéphir il faut utiliser les scripts `sauvegarde.sh` et `restauration.sh`.

1. Migration sans réinstallation vers EOLE 2.9

Le passage de la version mineure 2.8.1 vers une version 2.9 constitue un passage vers une version majeure. Le script `Upgrade-Auto` disponible sur le serveur permet d'effectuer manuellement la migration d'un module vers les dernières versions stables.

Utiliser un fichier meta-release-lts alternatif

Le fichier `meta-release-lts` contient les adresses de tous les dépôts Ubuntu.

Dans un environnement sans accès direct à internet, on peut définir, via ce fichier, l'emplacement interne des dépôts à utiliser par `Upgrade-Auto`.

Dans l'interface de configuration du module, en mode expert, aller dans l'onglet **Dépôt tiers**.



La variable `upgrade_auto_meta_release_repo` permet de définir où est téléchargé le fichier `meta-release-lts`.

Upgrade-Auto pour les applications tierces

Vous pouvez inclure l'upgrade d'applications annexes.

Pour cela, il vous faudra créer un fichier `.txt` à l'emplacement suivant `/usr/share/eole/upgrade/source.list.d/`.

Puis indiquer à l'intérieur la version cible (donc la version vers laquelle vous voulez faire l'upgrade), le dépôt actuel, et le nouveau dépôt (celui qui permet d'avoir la version souhaitée) séparés par des ";".

Exemple

Créer le fichier `/usr/share/eole/upgrade/source.list.d/mtes.txt` avec les lignes suivantes :

```
2.7.2;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu xenial
main;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu bionic
main
2.7.2;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu xenial
main;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu bionic
main
```

Puis lancer la commande `Upgrade-Auto`. Toutes les applications installées depuis le dépôt source seront mises à jour vers leur version du dépôt de destination.

Explication du contenu : 2.7.2 est la version vers laquelle nous souhaitons migrer, le premier ppa inscrit est celui qui est actuellement sur le serveur, le second est le ppa souhaité pour la version 2.7.2.

Mise à disposition

Le script `Upgrade-Auto` est disponible directement sur le serveur EOLE 2.8.1 après mise à jour de celui-ci.

Fonctionnement général

Le script `Upgrade-Auto` permet d'effectuer la migration de façon automatique.

Les étapes sont :

1. choix de la version EOLE 2.9 de destination ;
2. mise à jour des fichiers `sources.list` ;
3. téléchargement des paquets ;
4. installation des paquets ;
5. mise à jour automatique des fichiers de configuration (`*.eol`) ;
6. mise à jour des paramètres du serveur sur le serveur Zéphir si le serveur y est enregistré.

Procédure détaillée

Pré-requis

Par précaution, il est recommandé de faire une sauvegarde du serveur avant de lancer le script de migration.

La mise à jour du système n'est pas automatisée car elle est susceptible de poser des questions à l'utilisateur en cours d'exécution ;

DKMS

La procédure de migration refusera de s'exécuter si elle détecte des pilotes compilés (DKMS [p.40]).

Les DKMS sont en effet susceptibles de faire échouer la procédure : impossibilité de démarrer sur le nouveau noyau, fichier présent dans le paquet DKMS fourni par un autre paquet en standard...



L'utilisation du script de migration est critique, il n'y a pas de retour en arrière possible une fois les paquets installés.

Réalisation de la migration

1. Mettre à jour le serveur EOLE 2.8.1 :

```
# Maj-Auto
```

2. Reconfigurer et re-démarrer le serveur (si nécessaire) :

```
# reconfigure
```

```
# reboot
```

3. Lancer le script de migration :

```
# Upgrade-Auto
```

4. Convertir le fichier `config.eol` du module sauvegardé

- Si le serveur est enregistré sur un module Zéphir, il est possible de préparer la configuration et de la récupérer avec la commande `enregistrement_zephir` :

```
# enregistrement_zephir
```

- Sinon, exécuter l'interface de configuration du module, vérifier les valeurs et enregistrer.

5. Sur certains modules une mise à jour du serveur est requise avant de l'instancier, si cela demandé, il faut exécuter la commande suivante :

```
# Maj-Auto
```

6. Instancier et re-démarrer le serveur (si nécessaire) :

```
# instance
```

```
# reboot
```



Screen est une console virtuelle qui permet de détacher un terminal du processus appelant. Il permet notamment de récupérer la console virtuelle en cas de coupure du tunnel SSH.

Pour utiliser Screen :

```
# screen
```


Pour ouvrir une nouvelle console virtuelle :

`Ctrl + a` puis `c`

Pour passer d'une console à l'autre :

`Ctrl + a` + puis `barre espace`

Pour mettre Screen en tâche de fond :

```
# screen -D
```

Pour récupérer une session Screen :

```
# screen -x
```

2. Migration avec réinstallation vers EOLE 2.9

`migration29.sh` est le script de migration qui permet de passer des versions EOLE 2.7.2 ou 2.8.1 à la version 2.9 avec réinstallation du serveur.

Mise à disposition

Le script `migration29.sh` est mis à disposition sur le site FTP du projet EOLE.

Il est fonctionnel pour les modules 2.7.2 et 2.8.1 : Eolebase, Amon et Scribe ainsi que pour le module AmonEcole 2.8.1.

Fonctionnement général

L'utilisation d'un support externe est recommandé.

Le script permet de réaliser une sauvegarde puis une restauration spécifiques.

Des traitements sont effectués sur les données au cours des 2 opérations.

Procédure détaillée

Prérequis

Pour des raisons pratiques, la migration nécessite l'utilisation d'un support externe (partage distant ou disque USB) disposant d'un espace libre suffisant et d'un système de fichiers supportant les liens symboliques (ext3 conseillé).

Il est très fortement conseillé :

- de ne pas utiliser le même serveur que votre module 2.7/2.8 pour accueillir votre module EOLE 2.9 (ou tout du moins pas le même disque dur)
- de posséder une sauvegarde récente du serveur en version 2.7/2.8.

Le serveur 2.7/2.8 doit avoir été mis à jour, reconfiguré et redémarré si nécessaire.

Sauvegarde spéciale du module en version 2.7.2 ou 2.8.1

1. Télécharger le script de migration :

```
# wget ftp://eole.ac-dijon.fr/pub/Outils/migration/migration29.sh
```

2. Lancer la sauvegarde spécifique :

```
# sh migration29.sh
```

3. Éteindre ou débrancher le serveur du réseau

Restauration spéciale du module en version 2.9

1. Installer le module à partir de la dernière image EOLE 2.9

2. Mettre à jour le module :

```
# Maj-Auto
```

3. Si le module utilise les conteneurs, comme c'est le cas pour AmonEcole, il faut recréer les conteneurs :

```
# gen_conteneurs
```

4. Convertir le fichier `config.eol` du module sauvegardé

- Si le serveur est enregistré sur un module Zéphir, il est possible de préparer la configuration et de la récupérer avec la commande `enregistrement_zephir` :

```
# enregistrement_zephir
```

- Sinon, placer l'ancien fichier de configuration avec l'extension `.eol` dans le répertoire `/root` (pour faciliter sa sélection depuis le snap firefox, par exemple), exécuter la commande `gen_config` pour lancer l'interface de configuration du module, utiliser la fonctionnalité : **Fichier -> Importer une configuration**, vérifier les valeurs et enregistrer.

5. Instancier le serveur à l'aide du fichier converti :

```
# instance
```

6. Télécharger le script de migration :

```
# wget ftp://eole.ac-dijon.fr/pub/Outils/migration/migration29.sh
```

7. Lancer la restauration spécifique :

```
# sh migration29.sh
```

8. Reconfigurer le serveur :

```
# reconfigure
```



Auparavant, il était nécessaire de faire attention au répertoire de copie de l'ancien fichier de configuration pour l'étape de conversion. Les restrictions de droits pour l'application de configuration du module ne s'appliquent plus à partir de la version 2.9.0 du module.

Éléments migrés

Éléments communs

- les certificats SSL ;
- les personnalisations SSO ;
- l'adresse du serveur de test de bande passante ;
- les rôles personnalisés via l'EAD ;
- la désactivation de la mise à jour hebdomadaire.

Éléments spécifiques à Amon/AmonEcole

- Personnalisations e2guardian :
 - bases filtres activées ;

- sites, extensions et types mimes interdits ;
- sites autorisés ;
- mode de filtrage ;
- Personnalisations Squid :
 - domaines non mis en cache ;
 - domaines non soumis à l'authentification ;
- Personnalisations du routage :
 - groupes de machine ;
 - destinations interdites ;
 - sources interdites ;
 - règles optionnelles ;
 - horaires du pare-feu.

Éléments spécifiques à Scribe

- l'annuaire et le `.reader` ;
- l'annuaire Active Directory (selon version) ;
- les données liées à Samba et à CUPS ;
- les données (`/home`) ;
- les données du serveur Bcdi Web ;
- les boîtes aux lettres locales ;
- les listes de diffusion ;
- les quotas utilisateur ;
- les ACL sur les données ;
- certaines bases MySQL (sympa) ;
- la configuration Bareos ;
- la configuration WPKG ;
- les modèles de partage ;
- les adresses réservées dans le DHCP.

⚠ Applications web

La restauration des données liées aux applications web Envoile fera l'objet de scripts dédiés.

Suivi du script

Le script `migration29.sh` est versionné dans le dépôt Git Creole (branche *migration*) :

<http://dev-eole.ac-dijon.fr/gitweb/gitweb.cgi?p=creole.git;a=tree;h=refs/heads/migration;hb=refs/heads/mi>

Chapitre 5

Migration de 2.9.n vers 2.9.n+x : Maj-Release

Le passage d'une version 2.X.n vers 2.X.n+x comme par exemple de 2.7.1 à 2.7.2 équivaut à une conséquente mise à jour. Ce passage est manuel et volontaire et se fait à l'aide de la commande `Maj-Release`.

Le script `Maj-Release` disponible sur le serveur permet d'effectuer cette mise à jour.



Aucune réinstallation n'est nécessaire pour passer de la version 2.X.n vers 2.X.n+x.

Mise à disposition

L'ajout de nouvelles fonctionnalités entraîne une nouvelle version d'EOLE (2.9.n). Le passage d'une version mineure à une autre est manuel et volontaire.

La commande `Maj-Release` permet de passer à une version mineure plus récente.

Fonctionnement général

Le script `Maj-Release` permet d'effectuer la migration de façon automatique.

Les étapes sont :

- choix de la version EOLE 2.9 de destination ;
- mise à jour des fichiers `sources.list` ;
- téléchargement des paquets ;
- installation des paquets ;
- mise à jour automatique des fichiers de configuration (`*.eol`) ;
- mise à jour des paramètres du serveur sur le serveur Zéphir si le serveur y est enregistré.

Procédure détaillée

Pré-requis

Par précaution, il est recommandé de faire une sauvegarde du serveur avant de lancer le script de migration.

La mise à jour du système n'est pas automatisée car elle est susceptible de poser des questions à l'utilisateur en cours d'exécution ;



L'utilisation du script de migration est critique, il n'y a pas de retour en arrière possible une fois les paquets installés.

Réalisation de la migration

1. Mettre à jour le serveur EOLE 2.9.n :

```
# Maj-Auto
```

2. Reconfigurer et re-démarrer le serveur (si nécessaire) :

```
# reconfigure  
# reboot
```

3. Lancer le script de migration :

```
# Maj-Release
```

4. Reconfigurer le serveur :

```
# reconfigure
```



Screen est une console virtuelle qui permet de détacher un terminal du processus appelant. Il permet notamment de récupérer la console virtuelle en cas de coupure du tunnel SSH.

Pour utiliser Screen :

```
# screen
```

Pour ouvrir une nouvelle console virtuelle :

```
Ctrl + a puis c
```

Pour passer d'une console à l'autre :

```
Ctrl + a + puis barre espace
```

Pour mettre Screen en tâche de fond :

```
# screen -D
```

Pour récupérer une session Screen :

```
# screen -x
```

Chapitre 6

Migration vers le module Sphynx 2.9

1. Migration sans réinstallation vers EOLE 2.9

Le passage de la version mineure 2.8.1 vers une version 2.9 constitue un passage vers une version majeure. Le script `Upgrade-Auto` disponible sur le serveur permet d'effectuer manuellement la migration d'un module vers les dernières versions stables.

Utiliser un fichier meta-release-lts alternatif

Le fichier `meta-release-lts` contient les adresses de tous les dépôts Ubuntu.

Dans un environnement sans accès direct à internet, on peut définir, via ce fichier, l'emplacement interne des dépôts à utiliser par `Upgrade-Auto`.

Dans l'interface de configuration du module, en mode expert, aller dans l'onglet `Dépôt tiers`.



La variable `upgrade_auto_meta_release_repo` permet de définir où est téléchargé le fichier `meta-release-lts`.

Upgrade-Auto pour les applications tierces

Vous pouvez inclure l'upgrade d'applications annexes.

Pour cela, il vous faudra créer un fichier `.txt` à l'emplacement suivant `/usr/share/eole/upgrade/source.list.d/`.

Puis indiquer à l'intérieur la version cible (donc la version vers laquelle vous voulez faire l'upgrade), le dépôt actuel, et le nouveau dépôt (celui qui permet d'avoir la version souhaitée) séparés par des ";" :

Exemple

Créer le fichier `/usr/share/eole/upgrade/source.list.d/mtes.txt` avec les lignes suivantes :

```
2.7.2;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu xenial
main;deb http://ppa.launchpad.net/slimbook/slimbook/ubuntu bionic
main
2.7.2;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu xenial
main;deb http://ppa.launchpad.net/libreoffice/ppa/ubuntu bionic
main
```

Puis lancer la commande `Upgrade-Auto`. Toutes les applications installées depuis le dépôt source seront mises à jour vers leur version du dépôt de destination.

Explication du contenu : 2.7.2 est la version vers laquelle nous souhaitons migrer, le premier ppa inscrit est celui qui est actuellement sur le serveur, le second est le ppa souhaité pour la version 2.7.2.

Mise à disposition

Le script `Upgrade-Auto` est disponible directement sur le serveur EOLE 2.8.1 après mise à jour de celui-ci.

Fonctionnement général

Le script `Upgrade-Auto` permet d'effectuer la migration de façon automatique.

Les étapes sont :

1. choix de la version EOLE 2.9 de destination ;
2. mise à jour des fichiers `sources.list` ;
3. téléchargement des paquets ;
4. installation des paquets ;
5. mise à jour automatique des fichiers de configuration (`*.eol`) ;
6. mise à jour des paramètres du serveur sur le serveur Zéphir si le serveur y est enregistré.

Procédure détaillée

Pré-requis

Par précaution, il est recommandé de faire une sauvegarde du serveur avant de lancer le script de migration.

La mise à jour du système n'est pas automatisée car elle est susceptible de poser des questions à l'utilisateur en cours d'exécution ;

DKMS

La procédure de migration refusera de s'exécuter si elle détecte des pilotes compilés (DKMS [p.40]).

Les DKMS sont en effet susceptibles de faire échouer la procédure : impossibilité de démarrer sur le nouveau noyau, fichier présent dans le paquet DKMS fourni par un autre paquet en standard...



L'utilisation du script de migration est critique, il n'y a pas de retour en arrière possible une fois les paquets installés.

Réalisation de la migration

1. Mettre à jour le serveur EOLE 2.8.1 :

```
# Maj-Auto
```

2. Reconfigurer et re-démarrer le serveur (si nécessaire) :

```
# reconfigure
```

```
# reboot
```

3. Lancer le script de migration :

```
# Upgrade-Auto
```

4. Convertir le fichier `config.eol` du module sauvegardé

- Si le serveur est enregistré sur un module Zéphir, il est possible de préparer la configuration et de la

récupérer avec la commande `enregistrement_zephir` :

```
# enregistrement_zephir
```

- Sinon, exécuter l'interface de configuration du module, vérifier les valeurs et enregistrer.

5. Sur certains modules une mise à jour du serveur est requise avant de l'instancier, si cela demandé, il faut exécuter la commande suivante :

```
# Maj-Auto
```

6. Instancier et re-démarrer le serveur (si nécessaire) :

```
# instance
```

```
# reboot
```



Screen est une console virtuelle qui permet de détacher un terminal du processus appelant. Il permet notamment de récupérer la console virtuelle en cas de coupure du tunnel SSH.

Pour utiliser Screen :

```
# screen
```

Pour ouvrir une nouvelle console virtuelle :

```
Ctrl + a puis c
```

Pour passer d'une console à l'autre :

```
Ctrl + a + puis barre espace
```

Pour mettre Screen en tâche de fond :

```
# screen -D
```

Pour récupérer une session Screen :

```
# screen -x
```

2. Sauvegarde et restauration

La procédure de sauvegarde/restauration peut s'effectuer sur des modules de versions différentes.

Il est possible d'utiliser cette procédure pour migrer un serveur en changeant de machine physique.

L'archive créée par la sauvegarde comprend :

- la base ARV et la configuration IPsec ;
- tous les dictionnaires et patches ;
- le fichier de configuration `config.eol`.

Sauvegarde

Des scripts de sauvegarde et restauration des données sont intégrés au serveur Sphynx. Pour démarrer une sauvegarde, il faut exécuter la commande `sauvegarde.sh` sur le serveur. Il n'y a rien de particulier à faire dans cette procédure. Le fichier de sauvegarde créé est `'date du jour'.tar.gz`.

Les sauvegardes générées sont stockées localement dans le répertoire `/var/lib/sphynx_backups/`. Il est donc recommandé de sauvegarder ce répertoire avec un système de sauvegarde externe.

Exemple de sauvegarde

```
root@sphynx:~# sauvegarde.sh
Sauvegarde en cours, patientez ...
- base ARV + Strongswan
- clés de connexion
- configuration eole
tar: Suppression de « / » au début des noms des membres
Compression de l'archive...
Archive créée : /var/lib/sphynx_backups/17-03-2015.tar.gz
root@sphynx:~#
```

Restauration

Si la restauration a lieu sur un nouveau serveur instancié ou non, il faut copier l'archive `tar.gz` souhaitée dans le répertoire `/var/lib/sphynx_backups/` (Le créer si nécessaire).

La restauration se fait via le script `restauration.sh`. Une liste des sauvegardes présentes est affichée. Taper en toutes lettres le nom de la sauvegarde à restaurer (date de création de la sauvegarde).

Vous aurez le choix de restaurer ou non la base ARV qui permet de générer les configurations IPsec.

Une sauvegarde restaurée sur un serveur de même version restaurera toute l'archive.

Une sauvegarde restaurée sur un serveur de version différente ne restaurera pas les patches et dictionnaires.

Attention, toutes les données modifiées depuis la sauvegarde (modifications de la base ARV, configurations IPsec) seront perdues lors de la restauration.

Exemple de restauration

```
root@sphynx:~# restauration.sh
Utilitaire de restauration sphynx
!!Attention : toutes les modifications effectuées après la
sauvegarde restaurée seront perdues!! liste des sauvegardes
présentes :
17-03-2015
sauvegarde à restaurer (rien pour sortir): 17-03-2015
décompression en cours...
```

```

vérification des données...
Restaurer les bases ARV et la configuration Strongswan (o/n) ?
[non] : o
- bases ARV + Strongswan
Stopping strongSwan IPsec...
* Stopping Serveur ARV: arv
* waiting for process to die [OK]
- configuration eole
Reconfigurez le serveur après la fin de la restauration
Suppression du répertoire temporaire...
Système restauré
root@sphynx:~#


```

Suivant le cas de figure, serveur instancié ou non, exécuter la commande `instance` ou `reconfigure` .

3. Migration de 2.9.n vers 2.9.n+x : Maj-Release

Le passage d'une version 2.X.n vers 2.X.n+x comme par exemple de 2.7.1 à 2.7.2 équivaut à une conséquente mise à jour. Ce passage est manuel et volontaire et se fait à l'aide de la commande `Maj-Release` .

Le script `Maj-Release` disponible sur le serveur permet d'effectuer cette mise à jour.

—  Aucune réinstallation n'est nécessaire pour passer de la version 2.X.n vers 2.X.n+x.

Mise à disposition

L'ajout de nouvelles fonctionnalités entraîne une nouvelle version d'EOLE (2.9.n). Le passage d'une version mineure à une autre est manuel et volontaire.

La commande `Maj-Release` permet de passer à une version mineure plus récente.

Fonctionnement général

Le script `Maj-Release` permet d'effectuer la migration de façon automatique.

Les étapes sont :

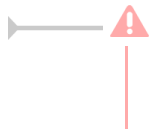
- choix de la version EOLE 2.9 de destination ;
- mise à jour des fichiers `sources.list` ;
- téléchargement des paquets ;
- installation des paquets ;
- mise à jour automatique des fichiers de configuration (`*.eol`) ;
- mise à jour des paramètres du serveur sur le serveur Zéphir si le serveur y est enregistré.

Procédure détaillée

Pré-requis

Par précaution, il est recommandé de faire une sauvegarde du serveur avant de lancer le script de migration.

La mise à jour du système n'est pas automatisée car elle est susceptible de poser des questions à l'utilisateur en cours d'exécution ;



L'utilisation du script de migration est critique, il n'y a pas de retour en arrière possible une fois les paquets installés.

Réalisation de la migration

1. Mettre à jour le serveur EOLE 2.9.n :

```
# Maj-Auto
```

2. Reconfigurer et re-démarrer le serveur (si nécessaire) :

```
# reconfigure
```

```
# reboot
```

3. Lancer le script de migration :

```
# Maj-Release
```

4. Reconfigurer le serveur :

```
# reconfigure
```



Screen est une console virtuelle qui permet de détacher un terminal du processus appelant. Il permet notamment de récupérer la console virtuelle en cas de coupure du tunnel SSH.

Pour utiliser Screen :

```
# screen
```

Pour ouvrir une nouvelle console virtuelle :

```
Ctrl + a puis c
```

Pour passer d'une console à l'autre :

```
Ctrl + a + puis barre espace
```

Pour mettre Screen en tâche de fond :

```
# screen -D
```

Pour récupérer une session Screen :

```
# screen -x
```

Chapitre 7

Migration vers le module Zéphir 2.9

Le serveur Zéphir est le premier serveur à faire migrer lorsqu'une nouvelle version de la distribution est disponible.

Zéphir gère les serveurs dont la version est inférieure ou égale à sa propre version.



- Les modules Zéphir 2.6 et 2.7 gèrent uniquement les modules de version supérieure ou égale à 2.4 et non supérieur à eux même, (zephir 2.7 ne gère pas les modules en 2.8 et 2.9)
- Les modules Zéphir 2.8 et 2.9 gèrent uniquement les modules de version supérieure ou égale à 2.6.0 et non supérieur à eux même, (zephir 2.8 ne gère pas les modules en 2.9)

Si la base de l'ancien serveur est importée dans la nouvelle version de Zéphir, tous les serveurs de version non gérée seront supprimés.

- Depuis Zéphir 2.7/2.8 vers Zéphir 2.9.n : Vérifier que le serveur est bien à jour en version stable, puis :
 - Exécuter le script `sauvegarde.sh` et mettre de côté l'archive générée ;
 - Installer la version voulue de Zéphir et effectuer une mise à jour stable (de préférence, conserver l'ancienne machine temporairement en cas de soucis) ;
 - Mettre en place l'archive créée précédemment dans `/var/lib/zephir_backups/` sur le nouveau serveur
 - Pour exécuter la restauration sur une nouvelle machine non instanciée il faut préciser le chemin absolu de la commande `/usr/share/eole/restauration.sh` (en effet les PATHS ne sont pas encore renseignés).
 - Éditer et sauvegarder la configuration avec `gen_config` et lancer `instance` (répondre non lorsqu'il est demandé de recréer les données).
- Depuis Zéphir 2.8.1 vers Zéphir 2.9 :
 - Utiliser le script `Upgrade-Auto` pour effectuer la mise à niveau du serveur (La configuration de Zéphir sera adaptée automatiquement si besoin) ;
 - reconfigurer le serveur.
- Depuis Zéphir 2.9.n vers Zéphir 2.9.n+x :
 - Utiliser le script `Maj-Release` pour effectuer la mise à niveau du serveur (La configuration de Zéphir sera adaptée automatiquement si besoin) ;
 - reconfigurer le serveur.

Chapitre 8

Gestion de la migration avec un serveur Zéphir

1. Généralités sur la migration

Le serveur Zéphir prend en compte la migration des serveurs enregistrés lors du passage à une nouvelle version de la distribution (suivant les cas supportés). Le but est de conserver l'identifiant du serveur en mettant à jour les informations sur la machine et en changeant le nom du module. Il propose également des fonctions permettant de préparer les configurations des serveurs à migrer lorsque cela est nécessaire.



La migration des données applicatives des serveurs enregistrés n'est pas gérée par le serveur Zéphir. La migration réelle du serveur est gérée par des scripts spécifiques à chaque module (si disponible) :

- `Upgrade-Auto` et `Maj-Release` (dans le cas où le serveur peut être mis à niveau par mise à jour) ;
- Scripts spécifiques de sauvegarde / restauration si une réinstallation est nécessaire (ex : `migration2X.sh`).

Une documentation spécifique est mise à disposition pour ces procédures lorsqu'elles deviennent disponibles.

Certaines données connues du serveur Zéphir (fichiers divers : voir le chapitre suivant pour les différents cas gérés) peuvent être récupérées automatiquement. Par contre, les modifications effectuées sur les serveurs (patches, dictionnaires locaux, ...) doivent être mises à jour manuellement (en re-crétant des variantes équivalentes pour la nouvelle version des modules).

La migration d'un serveur dans l'application Zéphir est effectuée lors de l'enregistrement d'un serveur réinstallé avec la nouvelle version sur son ancien identifiant ou en fin de procédure `Upgrade-Auto` (ou `Maj-Release`).

Lorsqu'un serveur est ré-enregistré, les clés de connexion de l'ancien serveur sont invalidées et les anciennes données sont sauvegardées dans un répertoire de sauvegarde (si nécessaire, éteindre l'ancien serveur pour éviter qu'il remonte des journaux systèmes d'échec de surveillance).



Dans le cas d'une migration nécessitant une préparation de configuration (voir chapitre suivant), il est possible de revenir dans la configuration d'origine pour remettre l'ancien serveur en service.

Retour en arrière après migration

Dans la page de description du serveur dans l'application web Zéphir, un bouton `retour en version XXX` permet de remettre en place les données du module dans sa version précédente.

Cette manipulation entraîne la perte des modifications effectuées depuis le passage sur la nouvelle version (données éditées sur Zéphir, ou remontées sur Zéphir par le serveur). Il faut recommencer la procédure d'enregistrement pour migrer à nouveau le serveur.

Migration du serveur Zéphir

Le serveur Zéphir est le premier serveur à faire migrer lorsqu'une nouvelle version de la distribution est disponible.

Zéphir gère les serveurs dont la version est inférieure ou égale à sa propre version.



- Les modules Zéphir 2.6 et 2.7 gèrent uniquement les modules de version supérieure ou égale à 2.4 et non supérieur à eux même, (zephir 2.7 ne gère pas les modules en 2.8 et 2.9)
- Les modules Zéphir 2.8 et 2.9 gèrent uniquement les modules de version supérieure ou égale à 2.6.0 et non supérieur à eux même, (zephir 2.8 ne gère pas les modules en 2.9)

Si la base de l'ancien serveur est importée dans la nouvelle version de Zéphir, tous les serveurs de version non gérée seront supprimés.

- Depuis Zéphir 2.7/2.8 vers Zéphir 2.9.n : Vérifier que le serveur est bien à jour en version stable, puis :
 - Exécuter le script `sauvegarde.sh` et mettre de côté l'archive générée ;
 - Installer la version voulue de Zéphir et effectuer une mise à jour stable (de préférence, conserver l'ancienne machine temporairement en cas de soucis) ;
 - Mettre en place l'archive créée précédemment dans `/var/lib/zephir_backups/` sur le nouveau serveur
 - Pour exécuter la restauration sur une nouvelle machine non instanciée il faut préciser le chemin absolu de la commande `/usr/share/eole/restauration.sh` (en effet les PATHS ne sont pas encore renseignés).
 - Éditer et sauvegarder la configuration avec `gen_config` et lancer `instance` (répondre non lorsqu'il est demandé de recréer les données).
- Depuis Zéphir 2.8.1 vers Zéphir 2.9 :
 - Utiliser le script `Upgrade-Auto` pour effectuer la mise à niveau du serveur (La configuration de Zéphir sera adaptée automatiquement si besoin) ;
 - reconfigurer le serveur.
- Depuis Zéphir 2.9.n vers Zéphir 2.9.n+x :

- Utiliser le script **Maj-Release** pour effectuer la mise à niveau du serveur (La configuration de Zéphir sera adaptée automatiquement si besoin) ;
- reconfigurer le serveur.

Migration des serveurs enregistrés auprès de Zéphir

Depuis EOLE 2.5, les variantes sont copiées automatiquement entre 2 versions successives (voir chapitre suivant) et les configurations des serveurs sont adaptées automatiquement lors de la mise à niveau.

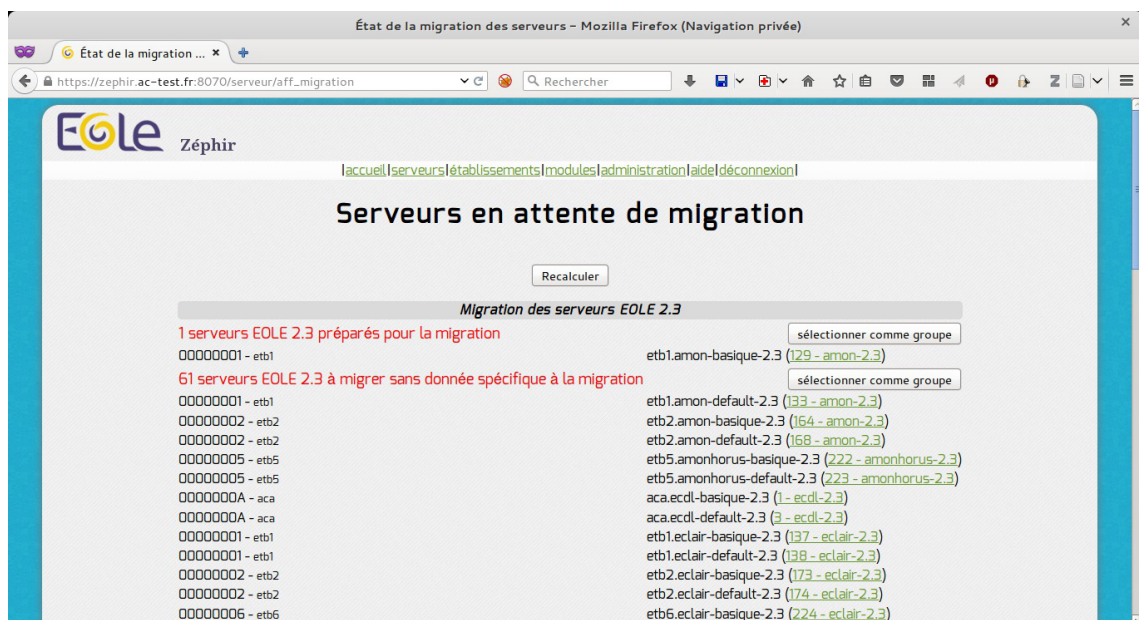
2. Préparation de la migration depuis l'application Zéphir

Page de suivi de la migration

Depuis la page d'accueil, un lien **suivi de la migration** affiche une page avec les informations suivantes pour chacune des versions d'EOLE utilisée sur les serveurs :

- la liste des serveurs à migrer sans données spécifiques à la migration ;
- la liste des serveurs préparés (dans l'application Zéphir) pour la migration.

Le bouton **sélectionner comme groupe** permet de créer un groupe contenant tous les serveurs de la liste.



Cette page concerne les migrations nécessitant de préparer la configuration des serveurs, pas les migrations entièrement automatisées (adaptation automatique du fichier de configuration par **Upgrade-Auto** sur la machine).

Préparation des variantes avant migration

La première tâche à prendre en compte pour préparer la migration est l'adaptation des variantes dans la nouvelle version. Suivant la version, deux cas sont possibles :

- adaptations nécessaires : pour les modules dont la configuration n'est pas directement compatible, les variantes du nouveau module doivent être créées manuellement dans l'application (il est toujours possible d'utiliser la fonction de copie de variante, sans garantie sur la compatibilité des fichiers transférés)
- pour les modules compatibles d'une version à l'autre (versions mineures successives), un bouton **Import des données X** est présent au niveau du menu des modules (par exemple, **Import des données 2.4.1** pour la distribution 2.4.2).



Cette fonction va effectuer les actions suivantes :

- Les dictionnaires locaux de la version X sont recopiés si ils n'existent pas ;
- les variantes n'existant pas (test sur le libellé) sont copiées et définies comme équivalentes (voir ci-dessous) ;
- les valeurs par défaut des modules X sont recopiées sur tous les modules où aucune n'est définie.

En cas de lancement successif de cette fonction, seuls les dictionnaires / variantes et fichiers de valeurs non présents dans la nouvelle version sont pris en compte. Cela évite d'écraser d'éventuelles adaptations faites entre-temps sur les nouvelles variantes. Cela implique qu'en cas d'ajout d'un fichier dans une variante X-1 déjà copiée, il faudra faire manuellement l'ajout dans la version X de la variante.

Liste des modules

Libellé	Identifiant	Nb de serveurs
EOLE-2.5.1 (Ubuntu trusty)		
Dictionnaires personnalisés Import des données 2.5.0		
Supprimer tous les modules		
amon-2.5.1	85	modifier supprimer variantes 0
amonecole-2.5.1	91	modifier supprimer variantes 0
eolebase-2.5.1	92	modifier supprimer variantes 0
horus-2.5.1	89	modifier supprimer variantes 0
scribe-2.5.1	84	modifier supprimer variantes 0
seshat-2.5.1	90	modifier supprimer variantes 0
sphynx-2.5.1	88	modifier supprimer variantes 0
thot-2.5.1	87	modifier supprimer variantes 0
EOLE-2.5.0 (Ubuntu trusty)		
Dictionnaires personnalisés		
amon-2.5.0	75	modifier supprimer variantes 0
amonecole-2.5.0	76	modifier supprimer variantes 0

Équivalence des variantes entre deux versions

Zéphir propose une notion d'équivalence de variantes entre deux versions successives de la distribution EOLE. Lorsque deux variantes sont définies comme équivalentes, Zéphir sera capable de choisir automatiquement la variante à attribuer au serveur lors de la procédure **Maj-Release**.

Les équivalences entre variantes peuvent être définies de deux façons :

- automatiquement, lorsque des variantes sont importées d'une version à l'autre de la distribution (voir ci-dessus). Dans ce cas, il est également possible de redéfinir manuellement ces équivalences.
- manuellement, en se rendant sur la page des variantes d'un module. Si la fonction est gérée pour cette version de la distribution, il sera possible de définir des équivalences pour chaque variante et chaque version de destination possible.

Liste des variantes relatives au module amon-2.4

Libellé	Clé Serveurs	Édition	équivalence 2.4.1	équivalence 2.4.2
test_11380 93	0	modifier supprimer importer/copier	test_11380 (94)	test_11380 (95)
standard 46	0		non modifiable	

[sauver ces correspondances](#)

[Ajouter une variante](#)

[Générer un rapport](#)

[Retour à la liste des modules](#)

Dans le cas d'une migration nécessitant une préparation de configuration, la variante à utiliser dans la nouvelle version est choisie au moment de générer la configuration, les équivalences ne sont donc pas prises en compte dans ce cas.

⚠ Saut de plusieurs versions

Il est possible dans certains cas de faire un saut de plusieurs versions (par exemple, Maj-Release de 2.6.0 vers 2.6.2).

Dans ce cas, il faut renseigner manuellement les équivalences des variantes depuis la page des variantes du module d'origine (par exemple, la page des variantes du module Amon

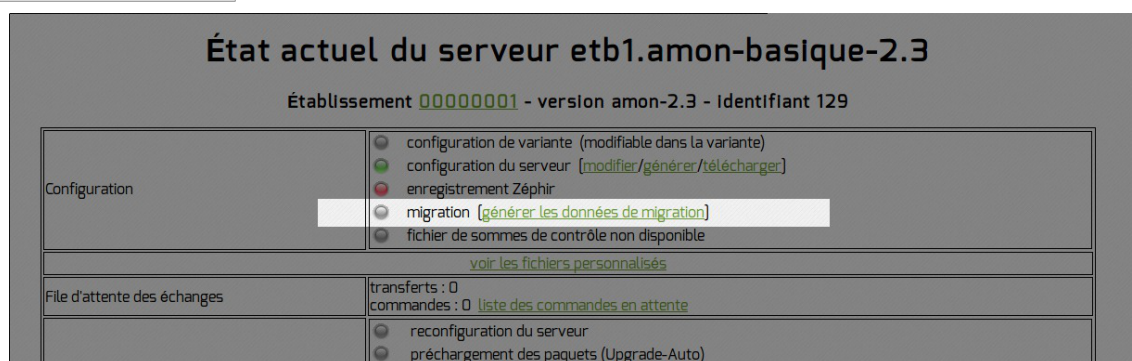
2.6.0). Zéphir définit les correspondances automatiquement, mais seulement lors de l'import des données entre deux versions mineures successives.

Cette étape est particulièrement importante si vous avez des variables supplémentaires définies dans des dictionnaires de variantes.

Générer une configuration de migration dans l'application web

Le serveur Zéphir permet de préparer à l'avance la configuration des serveurs à migrer. Cette fonction est disponible en cas de changement de version de la librairie Creole (nécessite d'adapter les dictionnaires personnalisés des variantes ou des serveurs), ou lorsque les fichiers de configuration sont trop différents pour être adaptés automatiquement d'une version à l'autre.

- sur la page d'état d'un serveur à migrer (amon-2.3 par exemple), cliquer sur le lien [générer les données de migration](#) ;



- dans la liste choisir la variante à utiliser lors de la migration du serveur et cliquer sur [Générer la configuration](#) ;

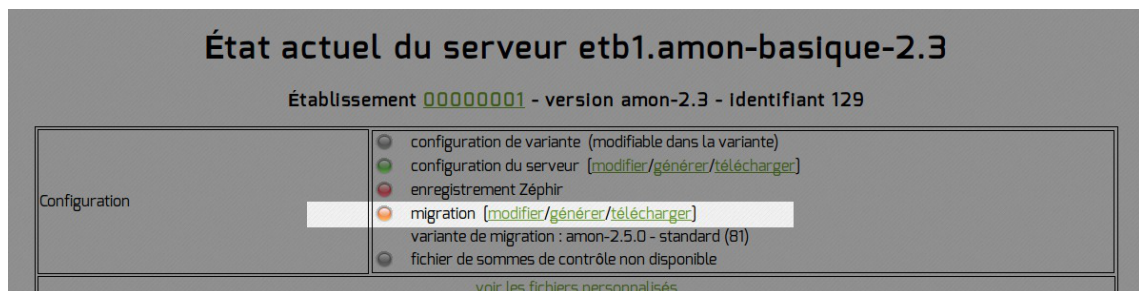


- pour les modules 2.4 et supérieurs, la saisie de configuration passe par l'interface de configuration du module qui a été intégrée au serveur Zéphir, le formulaire interne a été abandonné ;



- une fois tous les onglets renseignés, cliquez sur **Fichier** puis **Enregistrer la configuration**. Quitter l'interface de configuration du module. Un fichier **migration.eol** est alors enregistré dans le répertoire de données du serveur ainsi qu'un fichier **variante_migration** indiquant la variante sélectionnée.

Une fois la configuration enregistrée, vous pouvez choisir de modifier la configuration de migration ou de la re-générer en partant des valeurs par défaut (si vous voulez utiliser une autre variante par exemple). Les liens correspondants sont à côté du voyant migration en haut de la page d'état.



Cette étape de génération de configuration concerne seulement le fichier de configuration **config.eol** du serveur. Pour migrer les données applicatives (données de l'EAD, données des utilisateurs, ...), il faut passer par les scripts fournis pour sauvegarder/restaurer les données.

Si vous ne voulez pas passer par ces procédures, ou si vous avez des fichiers divers spécifiques à reprendre sur certains serveurs (hors variante), il est possible de définir une liste de fichiers à conserver sur Zéphir d'une version à l'autre (voir chapitre suivant).

Migration automatisée de fichiers de type 'fichiers divers'

Lors de la migration d'un serveur dans l'application, **les fichiers divers** ne sont pas repris automatiquement, aux exceptions suivantes près :

- les fichiers définis par défaut pour chaque module (configuration EAD, options de filtrage sur le serveur Amon et partages personnalisés sur les serveurs Horus / Scribe) sont remontés depuis le serveur migré en fin de procédure. Ce sont les scripts de migration (**migration2X.sh** ou **Upgrade-Auto**) qui se chargent de les restaurer sur la machine ;

- certains fichiers seront repris tels quels dans le répertoire du serveur sur le serveur Zéphir : configurations VPN sur sphynx / fichiers de réplication LDAP sur le module Seshat.

Pour d'autre cas (fichiers spécifiques à un serveur ou définis dans le fichier `fichiers_acad` d'une variante), il est possible de spécifier une liste de fichiers à remettre en place dans le répertoire `fichiers_zephir` du serveur après migration. Vous pouvez indiquer au serveur Zéphir la façon de les traiter en créant un fichier `/usr/share/zephir/migration_perso.py`. Les fichiers définis seront envoyés au serveur migré (avant la remontée des fichiers par défaut). Les données sont décrites sous la forme suivante :

```
# entête nécessaire pour que python sache comment traiter les accents
# -*- coding: UTF-8 -*-

# fichiers supplémentaires à transférer lors de la migration des données
migration_perso = {'amon-2.4':{'files':(['static-routes', 'static-routes', False]),
                                     'rights':(['static-routes', '', 'root', 'root', '0755']),
                                     'exclude':[]},
                  'horus-2.4': ...
                  }
# liste des destinations à ajouter dans fichiers_zephir
destinations_perso = {'amon-2.4':['/etc/sysconfig/static-routes'], 'horus-2.4':...}
```

exemple de migration d'un fichier static-routes vers amon-2.4



Le service `zephir` doit être relancé après ajout ou modification du fichier pour que celui-ci soit pris en compte.

La structure `migration_perso` décrit les fichiers à recopier dans le répertoire `fichiers_divers` après bascule sur la nouvelle version.

Le libellé complet du module de destination (nom-version) doit être spécifié, les fichiers pouvant ne plus être au même emplacement d'une version à l'autre de la distribution :

- `files` : source, destination, et booléen indiquant si le fichier doit être traduit de l'iso en utf-8 ;
- `rights` : fichier copié, options pour chmod / chown (par exemple, `-R` pour récursif), utilisateur, groupe, droits ;
- `exclude` : fichiers à exclure lors de la copie (utile si copie des répertoires entiers).

Les permissions définies dans la section `rights` sont automatiquement retranscrites dans le fichier `droits_zephir` du serveur (elles peuvent ensuite être modifiées / supprimées dans l'application web Zéphir si besoin). les chemins des fichiers source et destination sont donnés par rapport au répertoire `fichiers_zephir` du serveur sur le serveur Zéphir (pour référence : `/var/lib/zephir/conf/<id_etab>/<id_serveur>/fichiers_zephir/`)

- `destinations_perso` permet de renseigner la destination du fichier sur le serveur client. Cette liste de fichiers sera ajoutée dans le fichier `fichiers_zephir` du serveur (équivalent de la case `fichier de destination` dans la gestion des fichiers divers de l'application web Zéphir).



- Il est possible de mettre plusieurs entrées dans `migrations_perso` pour un même fichier de destination, seuls ceux retrouvés dans le répertoire du serveur seront recopiés (utile si un des fichiers a changé de nom entre EOLE 2.3 et EOLE 2.4 par exemple).

- Pour les destinations, elles seront toujours ajoutées au fichier `fichiers_zephir`, même si le fichier n'est pas présent. Dans ce cas, le fichier sera listé dans la section `fichiers absents` des fichiers personnalisés du serveur. Si besoin, Il est possible de les supprimer manuellement depuis l'application web Zéphir une fois le serveur migré.

Automatisation de la génération des configurations de migration

Il est possible de passer par l'API (XMLRPC) de l'application Zéphir pour générer en mode *batch* les configurations de migration. En particulier, dans le cas où les serveurs à migrer possèdent des variables supplémentaires définies dans des variantes.

Un squelette de script a été mis à disposition sur le serveur Zéphir pour aider à la migration vers EOLE 2.4 (`/usr/share/zephir/utils/gen_migration_sample.py`).

Le fonctionnement est le suivant :

- le script lit un fichier CSV indiquant les numéros de serveurs à préparer et le numéro de variante à leur appliquer une fois migrés ;
- pour chaque serveur, Il génère une configuration équivalente à celle obtenue via la fonction `générer les données de migration` dans l'application web Zéphir ;
- il passe ensuite dans une fonction `update_conf`, qui reçoit en paramètre la configuration actuelle du serveur, la configuration de migration générée et la variante de destination.

C'est cette fonction qu'il convient d'adapter pour remplir les informations manquantes dans la configuration de migration (voir les commentaires et exemples dans le script pour plus de détails).



Pour faciliter ce travail, il peut être intéressant de générer manuellement une première configuration depuis l'application web Zéphir afin de repérer les éventuelles données manquantes.

Dans le cas d'une migration vers EOLE 2.4, la nouvelle application de saisie de configuration offre quelques fonctions intéressantes :

- dans la liste de choix du mode (basique / normal / expert), il est possible d'activer le mode *debug* pour voir le nom des variables ;
- en cliquant sur `fichier` -> `enregistrer la configuration`, un tableau récapitulatif présente toutes les variables obligatoires non renseignées.

Une variable `debug` est présente dans le script, et permet d'afficher les valeurs de la configuration de migration sans la sauvegarder réellement sur le serveur Zéphir.

Il est aussi possible d'importer les fonctions dans un shell python pour récupérer les objets de configuration et faire des essais de manipulation. se placer dans le répertoire où se trouve le script et lancer l'invite python :

```
import gen_migration_sample
gen_migration_sample.debug = True
conf_actuelle, conf_migration =
gen_migration_sample('nom_du_fichier_CSV')
```

Le script contient également quelques fonctions utilitaires pour comparer / récupérer et

assigner des valeurs aux variables.

3. Migration après réinstallation d'un serveur

Cette procédure correspond au cas où la migration est effectuée suite à la réinstallation d'un serveur client. Cela est nécessaire dans certain cas :

- lorsque la migration n'est pas prise en compte par la distribution Ubuntu, ou que des modifications structurantes du système de fichiers sont nécessaires ;
- en cas de remplacement du matériel.

L'étape de migration des données du serveur est effectuée par un script effectuant une sauvegarde / restauration spécifiques à la migration (script `migration2X.sh`). Si le serveur n'a pas de données spécifiques à récupérer (ou uniquement des données gérées au niveau du serveur Zéphir), il est aussi possible de partir d'une installation vierge.

Une fois le serveur installé, la mise à jour du serveur dans l'application web Zéphir se fait par la procédure `enregistrement_zephir` :

- à la question `créer le serveur dans la base du serveur Zéphir ?` répondre `non` ;
- renseigner le numéro d'établissement (optionnel) et le n° identifiant l'ancien serveur ;
- un message d'avertissement indique que le serveur doit être migré dans l'application Zéphir, répondre `oui` ;
- différents cas sont possibles une fois arrivé au menu de finalisation :
 1. Une configuration a été préparée sur le serveur Zéphir (voir chapitres suivants) : utilisez le choix 2 → `Utiliser la configuration définie sur le serveur Zéphir`
 2. Aucune configuration n'est préparée, mais vous voulez récupérer des dictionnaires provenant d'une variante : modifier la variante (choix 4), puis utilisez le choix 2 → `Récupérer les fichiers de variante sur le serveur Zéphir`. Utiliser `gen_config` après l'enregistrement pour renseigner la configuration
 3. Vous avez généré la configuration (ou importé l'ancienne) avec l'outil `gen_config` local : utiliser le choix 3 → `Sauver la configuration actuelle sur le serveur Zéphir`
- si des fichiers divers ont été déclarés dans le fichier `migration_perso.py` (voir plus loin), répondez `oui` à la question `Voulez vous migrer ces données ?` ;
- dans le cas 1, suivez les instructions données en fin d'enregistrement pour instancier le serveur, dans les autres cas, lancez la procédure `instance` ;
- un fois le serveur instancié, lancer si besoin le script de restauration des données (ex : `migration2X.sh`) pour restaurer les données de l'ancien serveur.



À la fin de la procédure de restauration des données (`migration2X.sh`), une sauvegarde de configuration est lancée par le client Zéphir pour remonter d'éventuels fichiers restaurés (configuration EAD, configuration de filtrage locale, etc ...).

Si vous avez renseigné la configuration après l'enregistrement (cas 2) et que vous n'utilisez pas la restauration, pensez à effectuer l'une des deux actions suivantes après instance :

- lancer `/usr/share/zephir/scripts/zephir_client save_files` sur le serveur migré ;
- demander une sauvegarde de la configuration du serveur depuis l'application Zéphir.

4. Migration par mise à jour avec les procédures Upgrade-Auto / Maj-Release

Les procédures `Upgrade-Auto` et `Maj-Release` permettent de migrer un serveur existant vers une version supérieure sans réinstallation.

- `Upgrade-Auto` permet la migration entre 2 versions majeures d'EOLE
- `Maj-Release` entre 2 versions mineures (même version d'Ubuntu)



La version de l'outil Creole étant différente entre 2.3 et 2.4, cette migration nécessite une création manuelle des variantes dans la nouvelle version, et une génération de la configuration de migration pour chaque serveur (voir les chapitres précédents).

C'est également le cas pour chaque première version majeure de la distribution (Des différences importantes de version des logiciels de base pouvant impliquer des ajouts ou suppression de variables de configuration).

Le script `Upgrade-Auto` (ou `Maj-Release`) se charge de télécharger les nouveaux paquets et de mettre à jour le système vers sa nouvelle version.

Dans le cas d'un serveur enregistré sur le serveur Zéphir, la dernière étape de cette procédure consiste à mettre à jour les informations du serveur dans l'application Zéphir et à récupérer une éventuelle configuration préparée à l'avance sur le serveur Zéphir.

Le déroulement de cette étape est le suivant :

- saisie d'un compte valide (login / mot de passe) de l'application Zéphir.
L'utilisateur doit avoir les droits suivants :
 - lecture ;
 - actions sur les clients (avec ou sans modification de configuration) ou enregistrement ;
 - écriture sur les serveurs et les modules (ou Migration de serveur + Ecriture (modules)).
- Si la configuration n'a pas été préparée à l'avance, la liste des variantes disponibles est proposée ;
- Si des fichiers divers ont été déclarés dans le fichier `migration_perso.py`, la procédure propose de les récupérer ;
- Après descente de ces fichiers et des données de variante, une sauvegarde est effectuée pour remonter les fichiers locaux (configuration EAD,...) auprès du serveur Zéphir ;
- Suivre les indications données en fin de procédure pour finaliser la configuration du serveur.

Glossaire

DKMS = <i>Dynamic Kernel Module Support</i>	DKMS est un framework utilisé pour créer des modules noyau dont les sources ne résident pas dans celles du noyau Linux.
---	---