1. Mise à jour Open-Prod et des modules myfab

Procédure pour réaliser une mise à jour d'Open-Prod et ou des modules myfab.

- Présentation
- Connexion à l'environnement technique Linux
- Trouver et exécuter une commande myfab
- Configuration du Git
- Déroulement d'une mise à jour
- Installation d'un commit particulier
- Versions d'Open-Prod

Présentation

1. Préambule

Les solutions Open-Prod par myfab sont hébergées sur un environnement Ubuntu 18.04 / 20.04 / 22.04. Afin de maintenir en condition opérationnelle ou de mettre à jour votre ERP, myfab met à disposition un ensemble de commandes qui pourront être exécutées directement sur votre ou vos serveurs Ubuntu. Ces commandes seront disponibles en « ligne de commande ».

> **A noter que ces commandes devront nécessairement être exécutées sur le compte utilisateur avec lequel l'ERP a été installé.**

2. Mise à jour et versionning

L'ensemble des mises à jour d'Open-Prod et des modules myfab s'effectuent au travers d'un « Git ». Il s'agit d'un entrepôt de données sur lequel vos différents logiciels se connecteront afin de se mettre à jour : chaque connexion à un Git nécessitera une authentification spécifique qui aura été pré paramétrée lors de la mise en ligne de votre environnement. Chaque mise à jour logiciel devra s'accompagner d'une mise à niveau de la ou des bases de données existantes.

Connexion à l'environnement technique Linux

1. Connexion / Raccordement au(x) serveur(s)

Les connexions à l'environnement Open-Prod s'effectuent au travers du protocole "ssh". C'est un protocole permettant une connexion au(x) serveur(s) Ubuntu via une liaison sécurisée afin de pouvoir lancer les commandes nécessaires à la mise à jour / maintenance de l'environnement.

Il convient de détenir :

- L'adresse IP ou le nom de votre serveur,
- Le code utilisateur ayant réalisé l'installation d'Open-Prod sur le serveur,
- Le mot de passe du dit utilisateur.

Le poste à partir duquel la mise à jour sera réalisée devra être doté d'un client ssh, par exemple, Putty. Le poste devra également pouvoir joindre le serveur concerné sur le port 22.

> **Merci de vous rapprocher de votre administrateur pour vérifier si votre poste est équipé d'un tel outil et si le serveur est joignable à partir de ce dernier.**

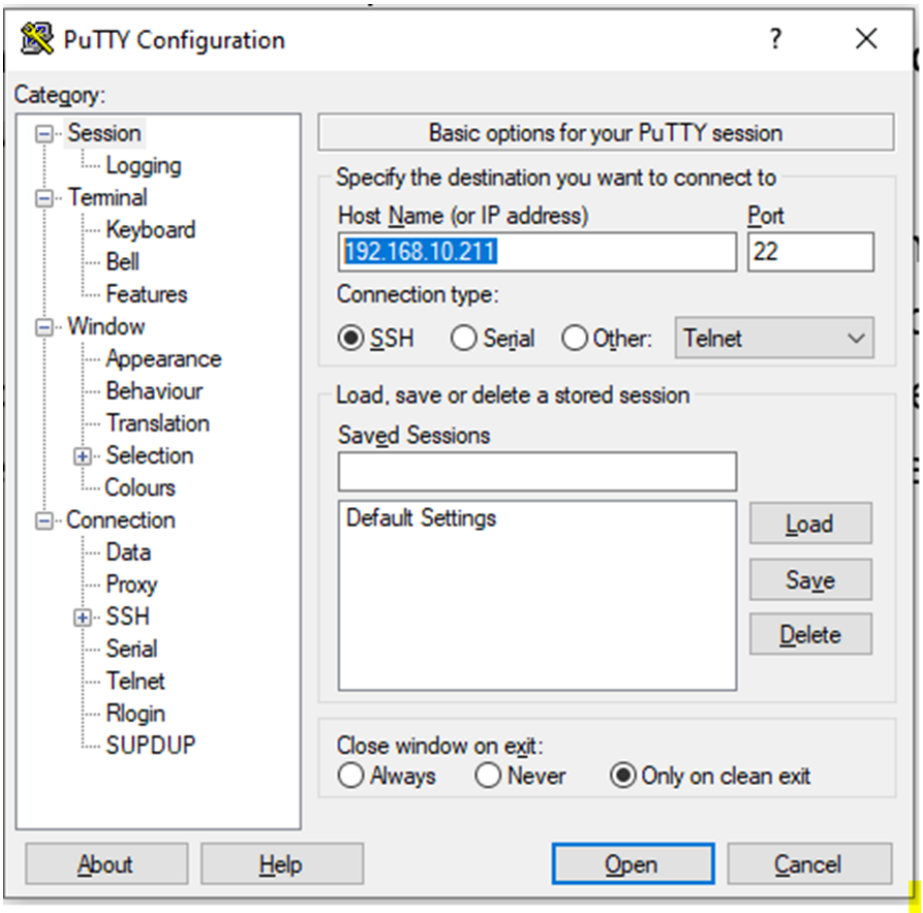
2. Mise en œuvre de Putty

Putty est téléchargeable à partir de l'url suivante :

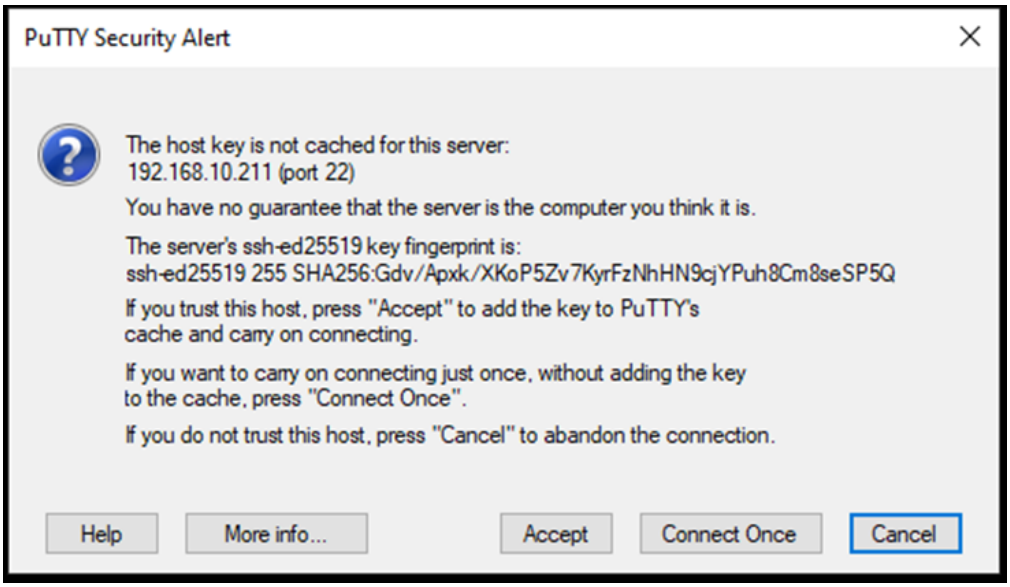
https://www.putty.org/

Télécharger et suivre les étapes d'installation de Putty, puis l'exécuter une fois cela fait.

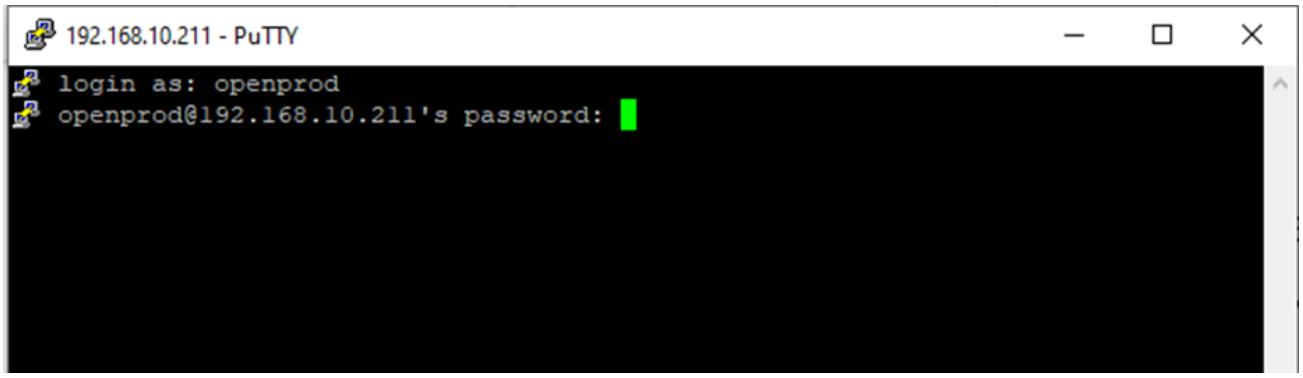
Saisir l'adresse IP du serveur à mettre à jour, puis cliquer sur « Open »



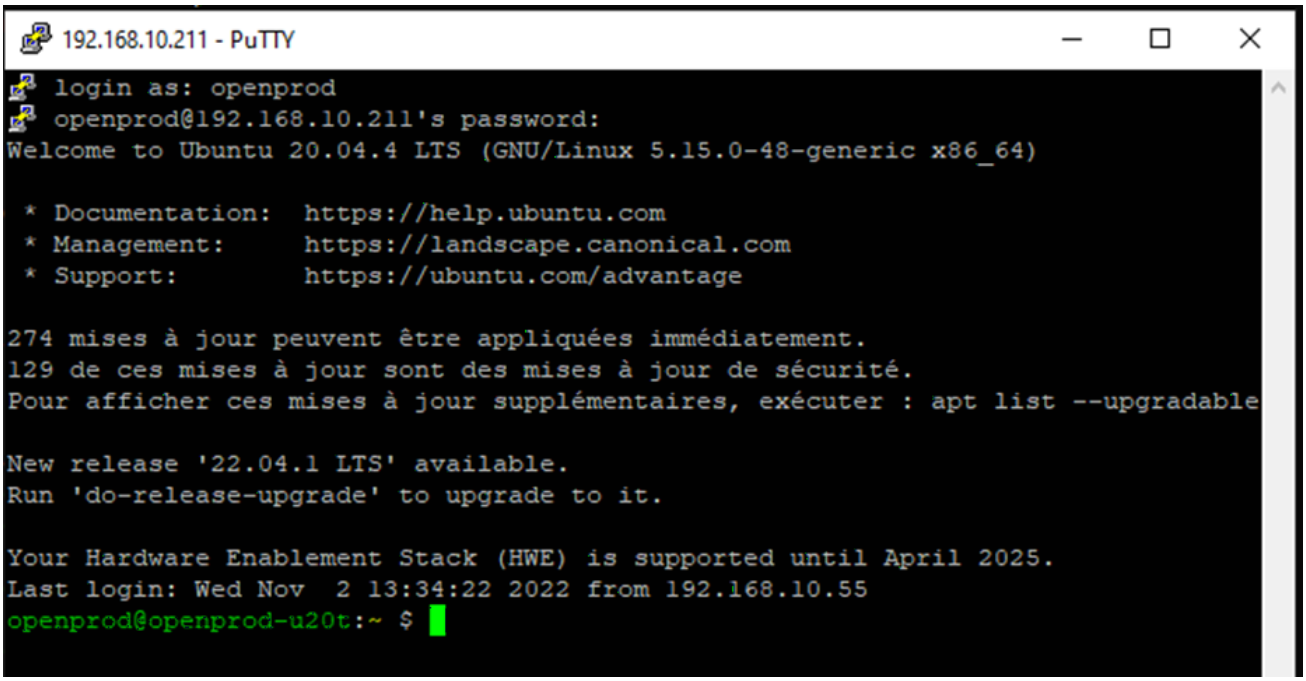
Votre pc peut afficher un message tel que celui ci-dessous lors de votre 1ère connexion sur le serveur. Cliquer sur « Accept ».



Enfin saisir le code utilisateur et le mot de passe lorsqu'ils vous seront demandés.



Si les identifiants de l'utilisateur sont corrects, l'écran ci-dessous doit s'afficher :



Le poste est à présent connecté en « ligne de commande » sur le serveur Ubuntu. L'utilisateur peut dorénavant taper les commandes d'administration.

3. Lecture des lignes de commandes

Une ligne est affichée à l'ouverture du terminal : avant le symbole $, diverses informations importantes en vert, et après le $, un curseur permettant la saisie de commandes.

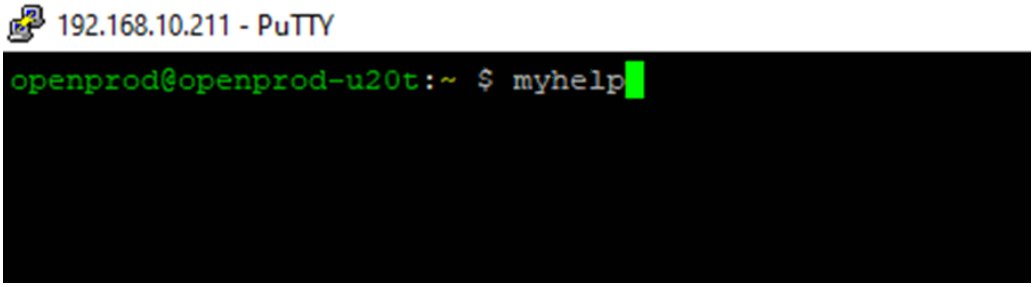
Les informations en vert se lisent ainsi :

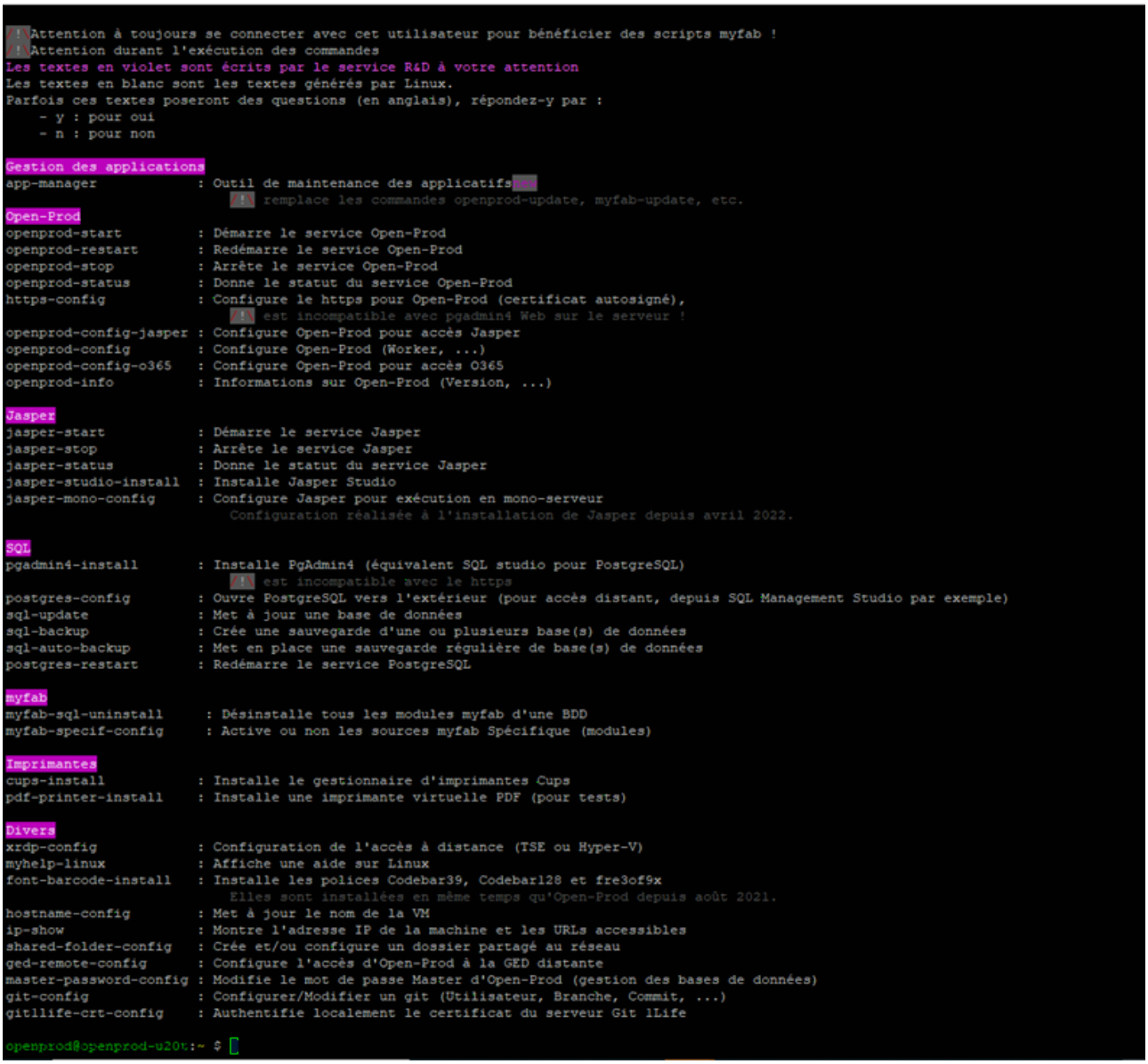
- **openprod** : nom de l'utilisateur actuellement connecté au terminal.
- **openprodmf** : nom de la machine sur laquelle est ouvert le terminal.
- ~ : emplacement actuel de la navigation du terminal dans l'arborescence de fichiers. Le symbole tilde ~ représente la Home de l'utilisateur courant (équivalent du C:/Users/utilisateur sur Windows).

Trouver et exécuter une commande myfab

Afin de simplifier la mise à jour et la maintenance des différents modules, 1Life a développé un ensemble de commandes mises à disposition de l’administrateur de l’ERP. Ces commandes sont recensées dans l'utilitaire « myhelp ».

Pour en afficher la liste, taper "`myhelp`" suivi de "Entrée" pour valider la commande.



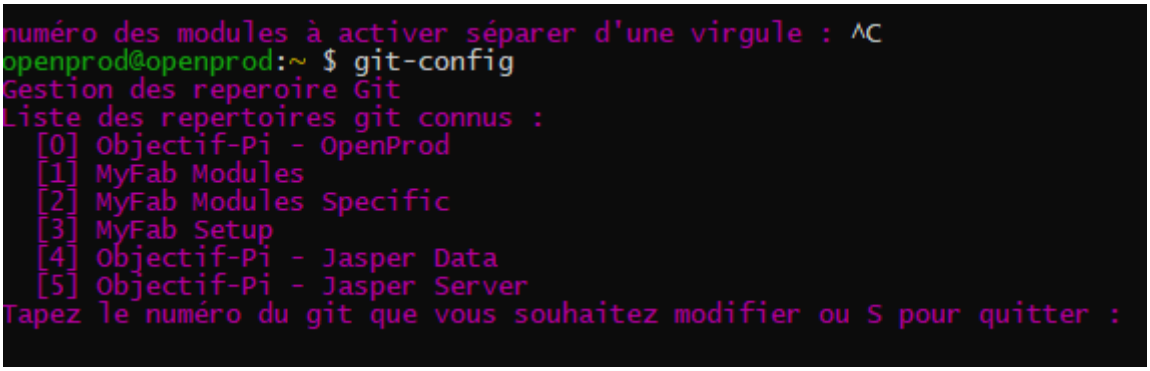


> Cette liste de commandes est maintenue et mise à jour en continu. **Pour plus de détails sur myhelp, se rendre sur la page dédiée.**

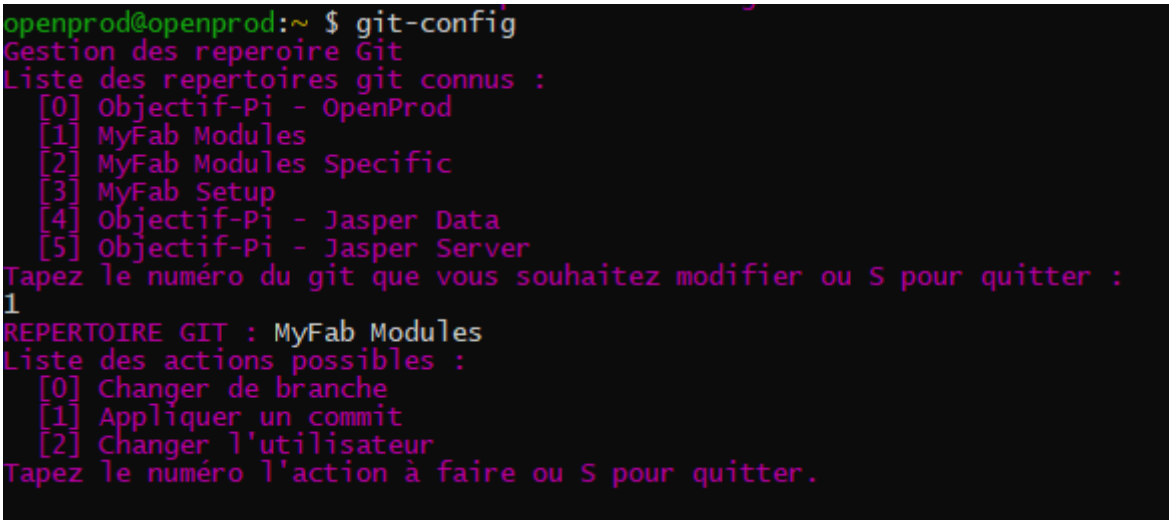
Configuration du Git

1. Commande git-config

Afin d'accéder aux différentes sources des programmes, il convient de paramétrer l'accès de l'utilisateur aux Gits Open-Prod et de myfab. Ce paramétrage est déjà en place sur un environnement Pour réaliser les opérations de maintenance courantes, vous aurez principalement besoin d'accéder aux Gits d'**Objectif-Pi Open-Prod**, **myfab Modules** et **myfab Modules Spécifiques**. Pour cela et depuis un terminal, utiliser la commande `git-config`



Saisir le numéro correspondant au Git que vous souhaitez modifier. Par exemple 1 pour le Git myfab.

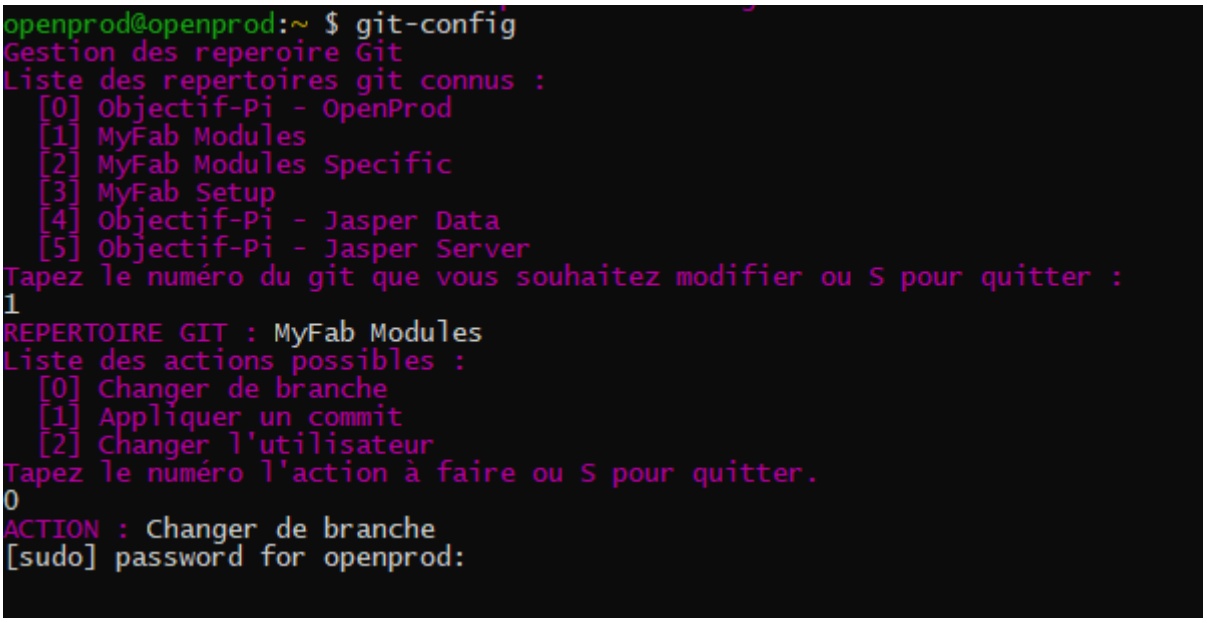


Trois options sont possibles :

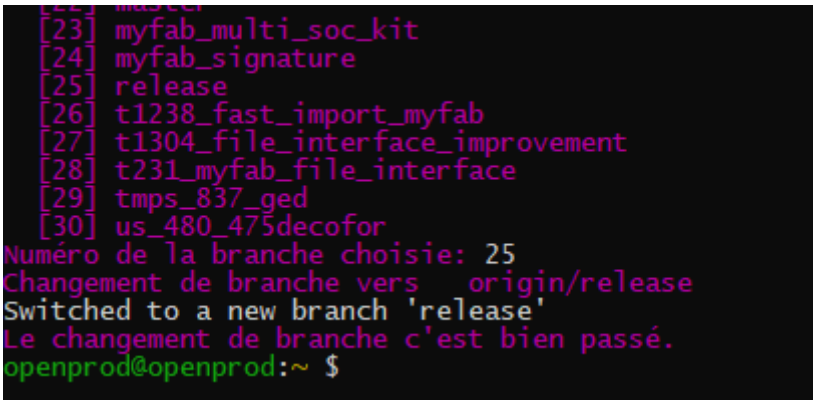
- Changer de branche [0],
- Appliquer un commit particulier [1],
- Changer d'utilisateur [2].

2. Changement de branche

Sur chacun des répertoires, il est possible de se positionner sur une branche du projet, c'est à dire une version. Après avoir tapé la commande `git-config` saisir 0 puis renseigner le mot de passe administrateur.

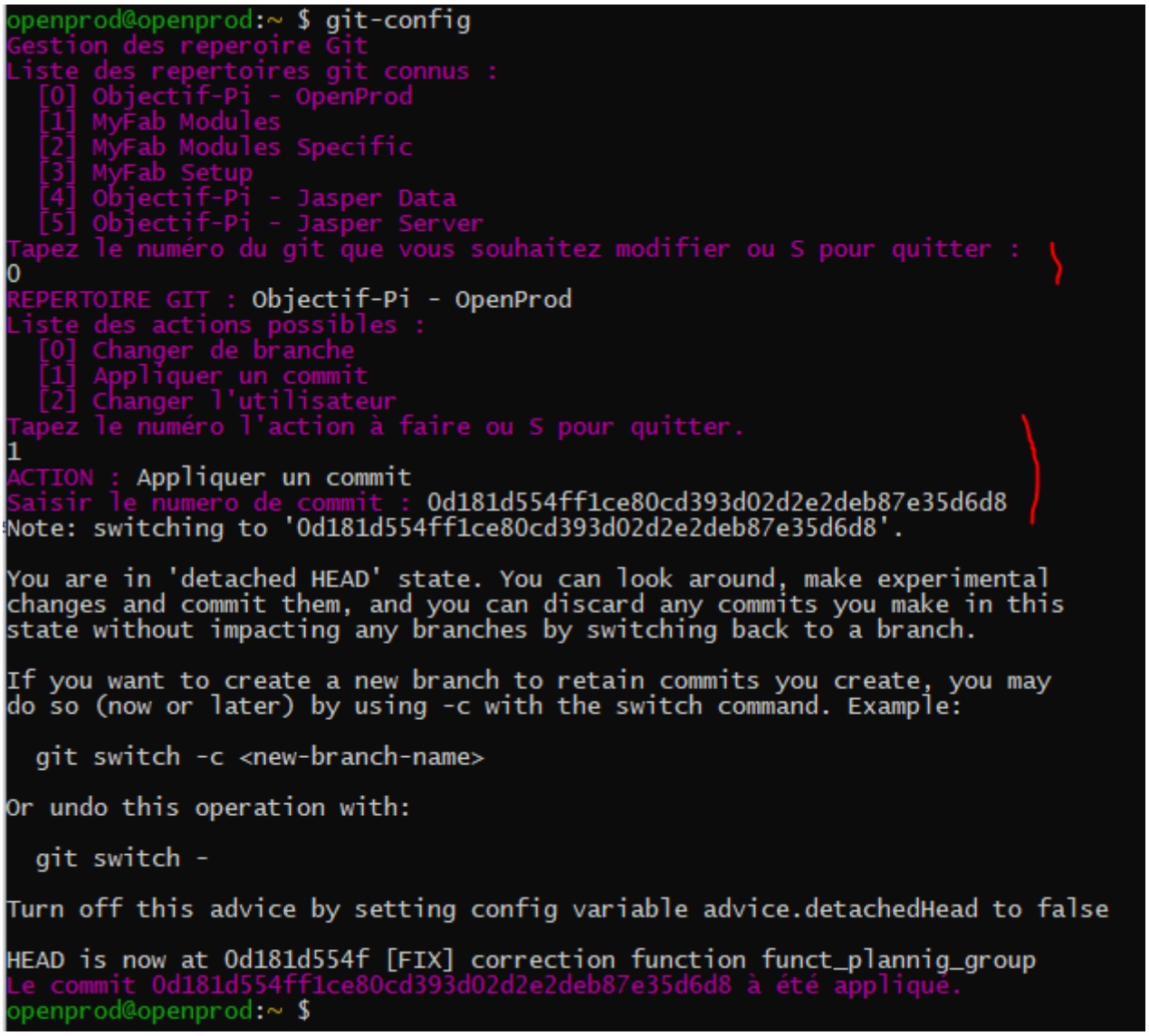


Toutes les branches mises en ligne sur le Git seront alors affichées. Taper le numéro correspondant à la branche sur laquelle vous souhaitez vous positionner.



3. Appliquer un commit particulier

Après avoir tapé la commande `git-config` et avoir sélectionné le Git à modifier, saisir 1 pour appliquer un commit donné. Saisir ensuite le numéro de commit puis valider par "Entrée".



4. Changer d'utilisateur

Après avoir tapé la commande `git-config` et avoir sélectionné le Git à modifier, saisir 2 pour modifier les identifiants d'accès au Git. Il vous sera demandé ensuite le login, l'email et le mot de passe de l'utilisateur à définir.

> Veuillez vous adresser au service SI ou à votre chargé de clientèle si vous ne connaissez pas ces identifiants.

```
openprod@openprod:~ $ git-config
Gestion des reperoire Git
Liste des repertoires git connus :
  [0] Objectif-Pi - OpenProd
  [1] MyFab Modules
  [2] MyFab Modules Specific
  [3] MyFab Setup
  [4] Objectif-Pi - Jasper Data
  [5] Objectif-Pi - Jasper Server
Tapez le numéro du git que vous souhaitez modifier ou S pour quitter :
2
REPERTOIRE GIT : MyFab Modules Specific
Liste des actions possibles :
  [0] Changer de branche
  [1] Appliquer un commit
  [2] Changer l'utilisateur
Tapez le numéro l'action à faire ou S pour quitter.
2
ACTION : Changer l'utilisateur
Repondez Oui (O) pour changer l'utilisateur actuel par un autre,
Repondez Non (N) pour supprimer l'utilisateur uniquement.
appuyez sur la touche O pour "Oui" ou sur la touche N pour "Non" :
O
Login de l'utilisateur MyFab Modules Specific : achapel
Email de achapel : achapel@1life.fr
Mot de passe de achapel :
```

Lors du changement d'utilisateur via la commande git-config, il peut arriver que la manipulation soit refusée par le système car le répertoire n'est pas considéré comme étant un répertoire "git" ou bien qu'il ne soit pas considéré comme "sûr".

```
openprod@openprod:~ $ git-config
Gestion des reperoire Git
Liste des repertoires git connus :
  [0] Objectif-Pi - OpenProd
  [1] MyFab Modules
  [2] MyFab Modules Specific
  [3] MyFab Setup
  [4] Objectif-Pi - Jasper Data
  [5] Objectif-Pi - Jasper Server
Tapez le numéro du git que vous souhaitez modifier ou S pour quitter :
2
REPERTOIRE GIT : MyFab Modules Specific
Liste des actions possibles :
  [0] Changer de branche
  [1] Appliquer un commit
  [2] Changer l'utilisateur
Tapez le numéro l'action à faire ou S pour quitter.
2
ACTION : Changer l'utilisateur
Repondez Oui (O) pour changer l'utilisateur actuel par un autre,
Repondez Non (N) pour supprimer l'utilisateur uniquement.
appuyez sur la touche O pour "Oui" ou sur la touche N pour "Non" :
O
Login de l'utilisateur MyFab Modules Specific : achapel
Email de achapel : achapel@1life.fr
Mot de passe de achapel :
fatal: not in a git directory
fatal: not in a git directory
fatal: not in a git directory
fatal: not in a git directory
fatal: unsafe repository ('/home/openprod/myfab-specific' is owned by someone else)
To add an exception for this directory, call:

        git config --global --add safe.directory /home/openprod/myfab-specific
```

Auquel cas, le terminal propose d'appliquer une exception sur le répertoire en question via une commande particulière :

```
git config --global --add safe.directory /home/openprod/myfab-specific
```

Il faut exécuter cette commande en ajoutant la commande `sudo` comme ci-dessous :

```
sudo git config --global --add safe.directory /home/openprod/myfab-specific
```

```
fatal: not in a git directory
fatal: unsafe repository ('/home/openprod/myfab-specific' is owned by someone else)
To add an exception for this directory, call:

        git config --global --add safe.directory /home/openprod/myfab-specific
openprod@openprod:~ $ sudo git config --global --add safe.directory /home/openprod/myfab-specific
openprod@openprod:~ $ git-config
Gestion des reperoire Git
Liste des repertoires git connus :
  [0] Objectif-Pi - OpenProd
  [1] MyFab Modules
  [2] MyFab Modules Specific
  [3] MyFab Setup
  [4] Objectif-Pi - Jasper Data
  [5] Objectif-Pi - Jasper Server
Tapez le numéro du git que vous souhaitez modifier ou S pour quitter :
2
REPERTOIRE GIT : MyFab Modules Specific
Liste des actions possibles :
  [0] Changer de branche
  [1] Appliquer un commit
  [2] Changer l'utilisateur
Tapez le numéro l'action à faire ou S pour quitter.
2
ACTION : Changer l'utilisateur
Repondez Oui (O) pour changer l'utilisateur actuel par un autre,
Repondez Non (N) pour supprimer l'utilisateur uniquement.
appuyez sur la touche O pour "Oui" ou sur la touche N pour "Non" :
O
Login de l'utilisateur MyFab Modules Specific : achapel
Email de achapel : achapel@1life.fr
Mot de passe de achapel :
openprod@openprod:~ $
```

Déroulement d'une mise à jour

Pour réaliser une mise à jour de l'environnement Open-Prod, veuillez suivre les étapes listées ci-dessous.

> **La commande `myhelp` est un récapitulatif de toutes les commandes créées par myfab pour faciliter l'accessibilité de l'interface Linux.**

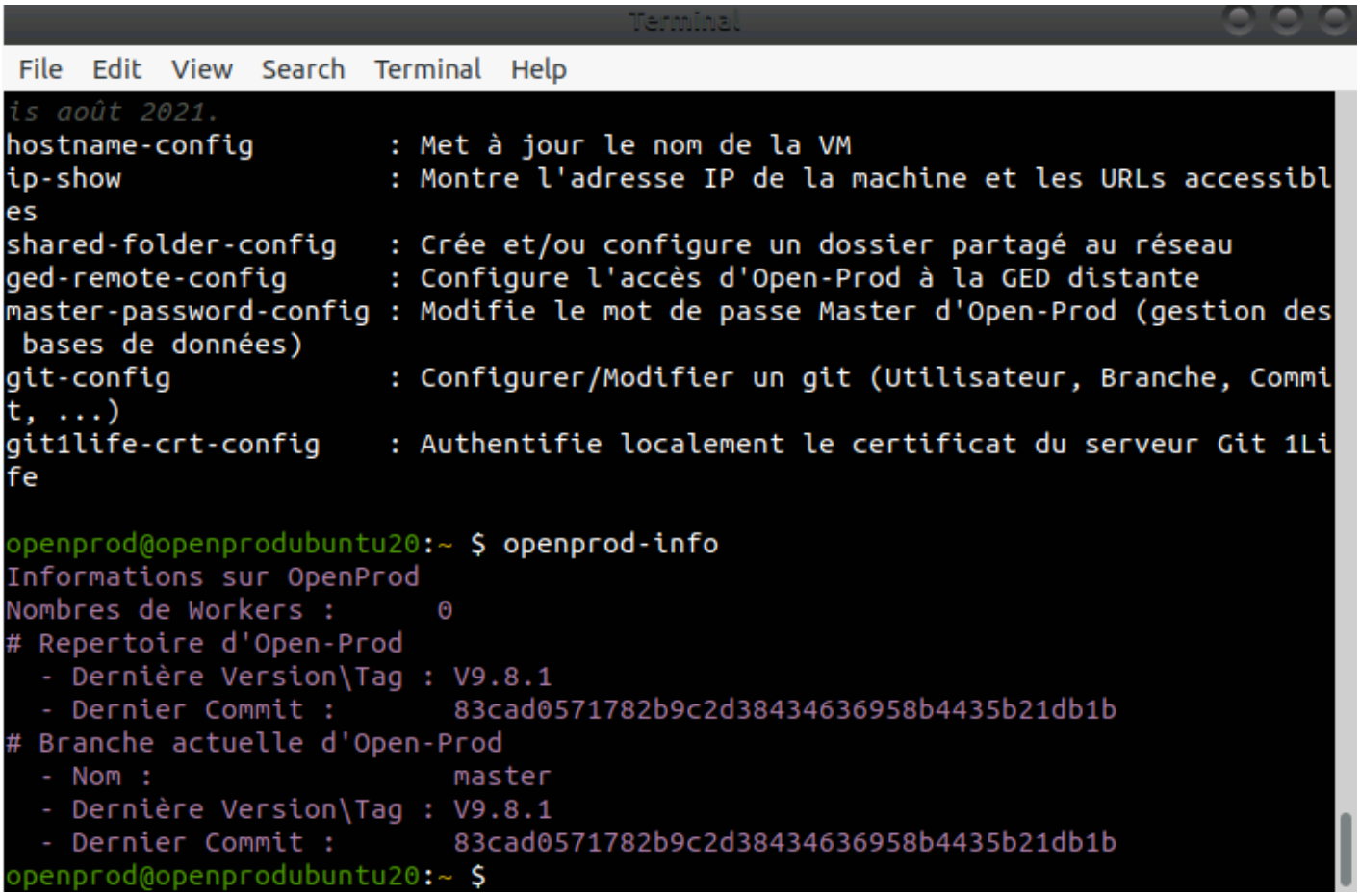
1. Réaliser une sauvegarde de l'environnement

> **Une mise à jour de votre environnement nécessite un arrêt de production durant toute la durée du traitement.**

> De plus, avant de faire une mise à jour, il est obligatoire de sauvegarder à minima les bases de données Open-Prod et Jasper à l'aide de la commande `sql-backup`.

> Si votre environnement est virtualisé, il est vivement conseillé de réaliser un snapshot de l'ensemble des VM concernées (sauvegarde de l'ensemble de l'environnement virtualisé avec les outils de l'hyperviseur). Si une régression était constatée à posteriori, il serait beaucoup plus simple de revenir en arrière par ce biais.

Il convient de noter et conserver le numéro de version d'Open-Prod avant la montée de version afin de savoir vers quelle version revenir au besoin. Pour cela, exécuter la commande `openprod-info`.



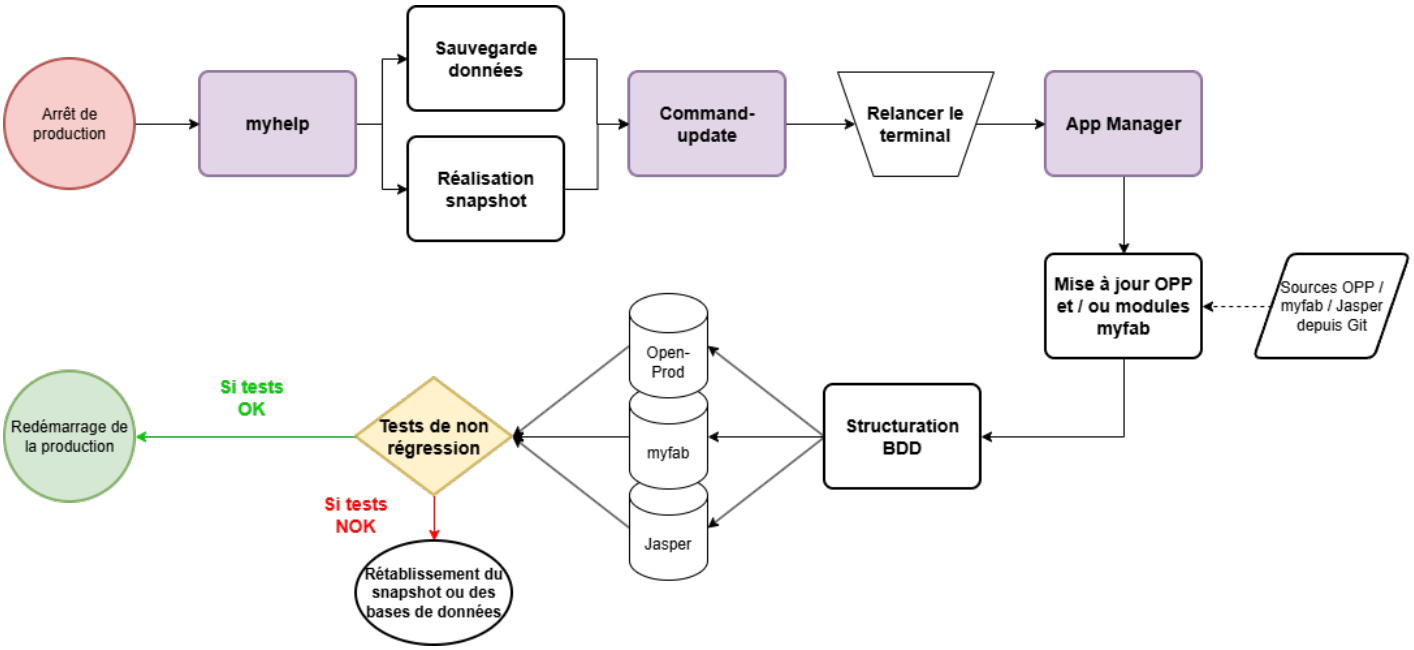
2. Mise à jour des scripts myfab

La prochaine étape consiste à mettre à jour les scripts myfab avec la commande `command-update`. Elle permet de mettre vos commandes Linux à jour de celles récemment ajoutées. Il est nécessaire de redémarrer le terminal après l'opération pour qu'il prenne en compte ces nouvelles commandes.

3. Lancement de l'App Manager

L'App Manager est le gestionnaire proposé par myfab pour gérer de manière simple l'installation et la mise à jour des applicatifs. Il permet d'exécuter de manière optimale les différentes commandes de mise à jour.

Résumé des étapes d'une mise à jour d'un environnement Open-Prod



Voir utilisation de **L'App Manager**

Installation d'un commit particulier

Il n'est pas systématiquement conseillé d'appliquer la dernière mise à jour disponible d'Open-Prod. Il est donc toujours possible de l'installer en sélectionnant un commit particulier.

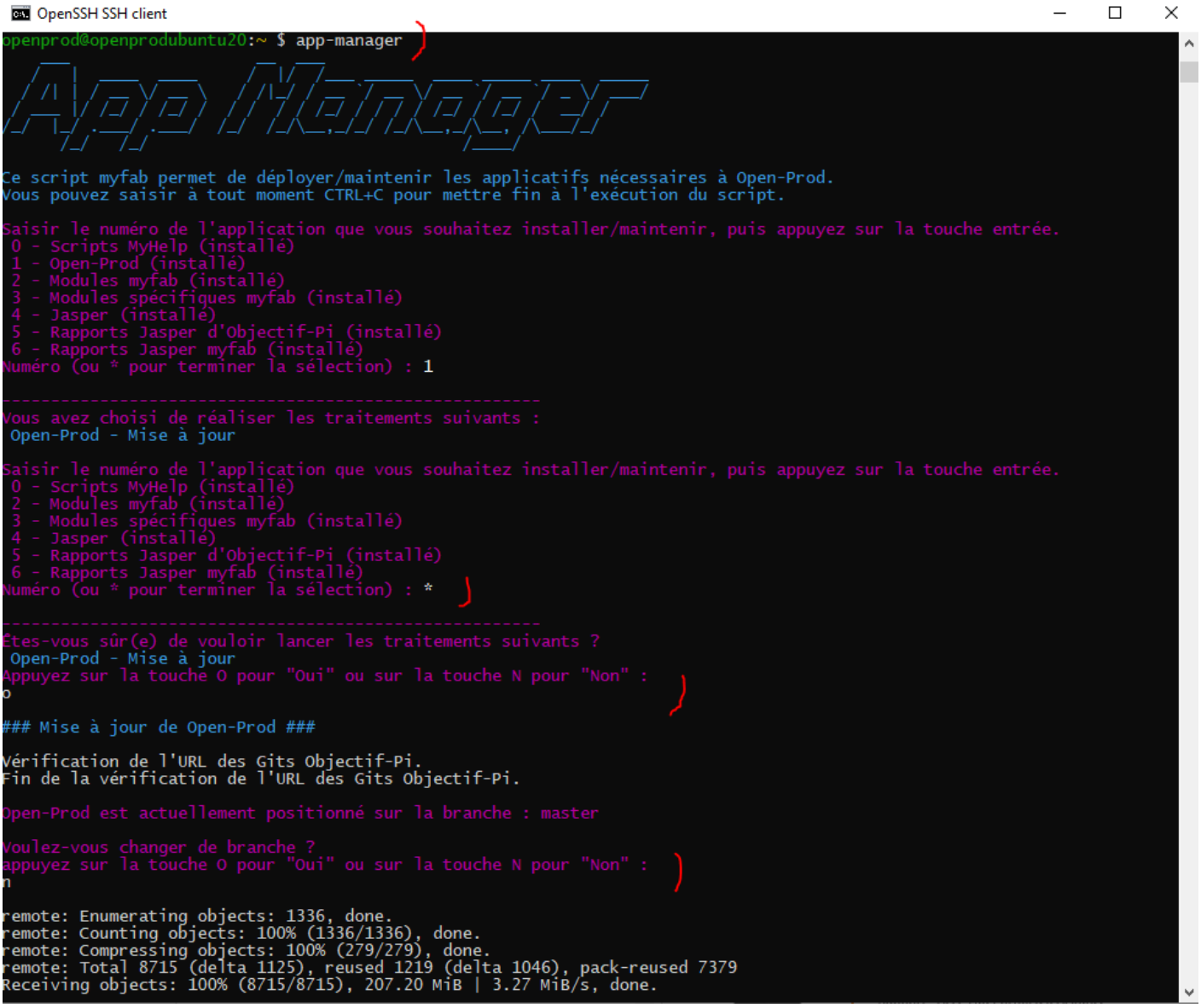
> Comme pour une mise à jour standard d'Open-Prod ou des modules myfab, il est vivement conseillé de réaliser une sauvegarde de votre environnement afin d'être en mesure de facilement revenir en arrière en cas de régressions.

1. Mise à jour d'Open-Prod sur un commit particulier

Les versions d'Open-Prod sont déclarées au travers du Git grâce aux commits. On pourrait décrire simplement un commit comme l'enregistrement d'une transaction en base de données. En l'occurrence, la transaction est la mise à disposition sur le git d'une version de l'ERP.

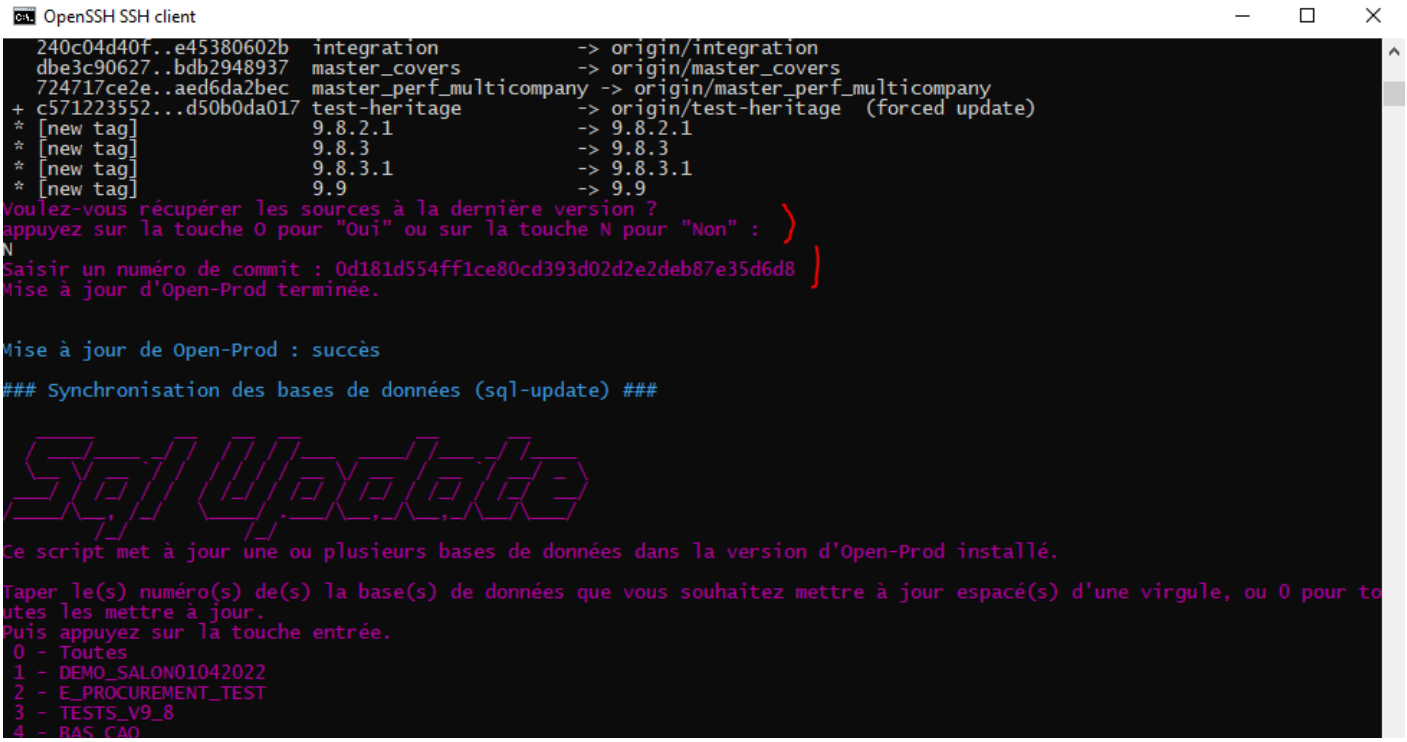
Pour appliquer un commit particulier d'Open-Prod suivre les étapes suivantes :

1. Lancer l'App Manager avec la commande `app-manager`.
2. Sélectionner l'option "1 - Open-Prod" et terminer la sélection avec la touche *. Puis valider la sélection avec la touche "O" pour Oui.
3. Le script va vérifier la connexion au Git d'Objectif Pi et le positionnement de branche. Il sera demandé si vous souhaitez changer de branche. Répondre par "N" pour Non. Le script va lancer la récupération des sources.



4. A la question "Voulez-vous récupérer les sources à la dernière version ?", répondre "N".

5. Saisir le numéro de commit souhaité.



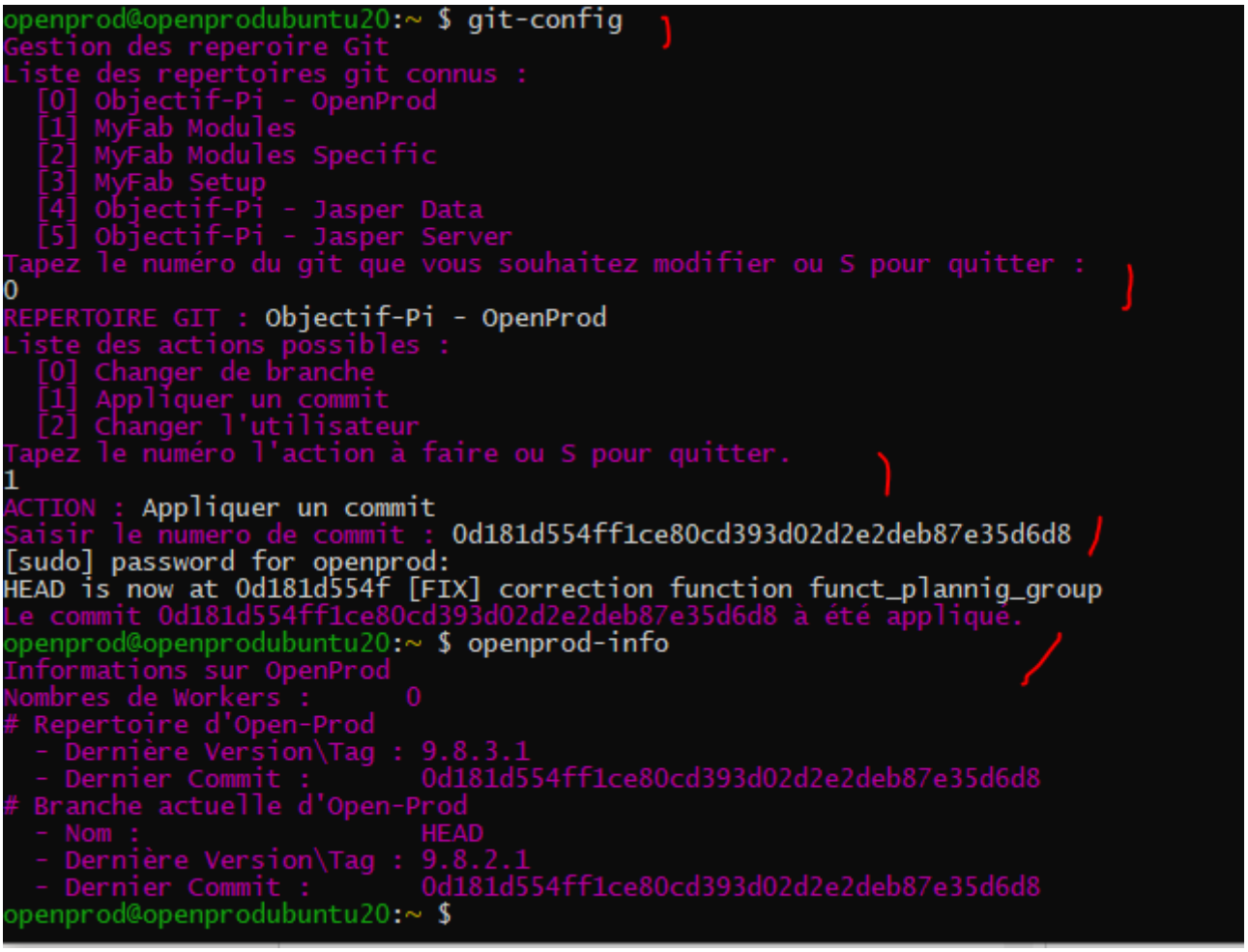
6. Réaliser la synchronisation SQL Update pour mettre à niveau les bases de données.

Après réalisation de ces étapes, vous pouvez vérifier la bonne application du numéro de commit en tapant la commande `openprod-info`. Si le commit installé n'est pas celui que vous avez tenté d'appliquer, positionner manuellement l'environnement sur un commit donné en suivant les étapes de la partie suivante.

2. Positionnement manuel sur un commit particulier

Une fois les sources récupérées, il est possible de se positionner manuellement sur un commit particulier à l'aide de la commande `git-config`. Procéder en suivant les étapes suivantes :

1. Depuis un terminal de commande, taper la commande `git-config`.
2. Sélectionner le répertoire Git Objectif Pi - Open-Prod en tapant "0".
3. Sélectionner l'action "[1] Appliquer un commit" en tapant 1.
4. Saisir le commit voulu et valider.
5. Saisir le mot de passe si demandé.
6. Lancer une synchronisation de votre / vos base(s) de données avec la commande `sql-update`.
7. Une fois le commit appliqué et votre / vos base(s) synchronisée(s), vous pouvez vérifier que la commande ait fonctionné en tapant la commande `openprod-info`.


openprod@openprodubuntu20:~ $ git-config
Gestion des reperoire Git
Liste des repertoires git connus :
[0] Objectif-Pi - OpenProd
[1] MyFab Modules
[2] MyFab Modules Specific
[3] MyFab Setup
[4] Objectif-Pi - Jasper Data
[5] Objectif-Pi - Jasper Server
Tapez le numéro du git que vous souhaitez modifier ou S pour quitter :
0
REPERTOIRE GIT : Objectif-Pi - OpenProd
Liste des actions possibles :
[0] Changer de branche
[1] Appliquer un commit
[2] Changer l'utilisateur
Tapez le numéro l'action à faire ou S pour quitter.
1
ACTION : Appliquer un commit
Saisir le numero de commit : 0d181d554ff1ce80cd393d02d2e2deb87e35d6d8
[sudo] password for openprod:
HEAD is now at 0d181d554f [FIX] correction function funct_plannig_group
Le commit 0d181d554ff1ce80cd393d02d2e2deb87e35d6d8 à été applique.
openprod@openprodubuntu20:~ $ openprod-info
Informations sur OpenProd
Nombres de Workers : 0
Repertoire d'Open-Prod
- Dernière Version\Tag : 9.8.3.1
- Dernier Commit : 0d181d554ff1ce80cd393d02d2e2deb87e35d6d8
Branche actuelle d'Open-Prod
- Nom : HEAD
- Dernière Version\Tag : 9.8.2.1
- Dernier Commit : 0d181d554ff1ce80cd393d02d2e2deb87e35d6d8
openprod@openprodubuntu20:~ $

Versions d'Open-Prod

> **Version majeure en diffusion d'Open-Prod et disponible sur la branche master : 9.10.**

Vous retrouvez ci-dessous la liste des versions majeures et le dernier commit associé. Ainsi, vous pouvez maintenir votre environnement dans une version majeure précise de l'ERP.

Dernière version majeure d'Open-Prod	Commit associé	Date
9.9	4a0fbcf0298f5bb57d9b10a1390431c21f4f0544	23/12/2023
9.9.1	a9ef3f26fca465cbeadbff9fe2b3ee0d701e1be0	19/05/2023