

Procédure d'upgrade depuis toute version antérieure vers V4.0 Rocky

A. Le plus facile, l'interface web

Il y a plusieurs cas, différenciés entre autres par le fait que vous ayez ou non un historique de connexions issu de versions *antérieures* à la V3.0 dans votre base de données. Donc concentrez-vous, décontractez les épaules et choisissez bien...

Cas A1. J'upgrade depuis V3.0 ou V3.1 *qui était ma première install de QoQ-CoT*** => je n'ai pas de connexions collectées par une version < V3.0 dans ma BD ***OU BIEN*** j'upgrade depuis V3.1 ***issue d'un upgrade depuis une version précédente*****

1. Pour simplifier ici la lecture, on nommera la base mysql QoQCoT (*pas de tirets « - » autorisés dans les noms de bases MySQL...*) et l'utilisateur associé QoQCoT_user.
2. Détarer la précieuse archive V4.0 Rocky, par exemple dans /tmp
3. Copier votre qoq-cot/config.php actuel dans /tmp/qoq-cot/src/
4. Vos données ne risquent rien, bien sûr, lors de cet upgrade... mais sauvez-les quand même... Dumpez donc soigneusement votre base QoQCoT.

5. UTILE UNIQUEMENT SI VOUS ÉTIEZ EN 3.0 = INUTILE SI VOUS ÉTIEZ EN 3.1 Modifiez votre base de données pour la rendre v3.1 compatible. Pour vous éviter de trop toucher à ces rouages internes recouverts de graisse forcément salissante, car nous tenons à votre confort, nous avons regroupé et automatisé toutes ces modifs dans le seul script /tmp/qoq-cot/ressources/UPGRADE_FROM_PREVIOUS_VERSIONS/ModifieDBV2toV3.sh.
Hmm, v2toV3 me dites-vous ? Mais je n'ai jamais eu de v2 !!!
Oui, c'est vrai... mais vous allez quand même devoir lancer le script pour certaines de ses actions qui corrigent quelques erreurs de jeunesse de la v3.0...

Mais quelles sont donc ces fameuses actions, me direz-vous, circonspect et méfiant comme doit l'être à raison un informaticien à l'heure du big data et de wikileaks ? Détaillons donc ça ensemble :

- a) **NORMALEMENT DÉJÀ FAIT LORS DE L'INSTALL de la V3.0** – Pour sécuriser l'insertion des données de connexion, QoQ-CoT depuis la version 3.0 Commandant Poulard préconise l'ajout d'un trigger dans la base de données, qui interdira toute possibilité (forcément malicieuse) de mettre à jour l'heure de fin d'une connexion avec une heure antérieure à celle déjà inscrite dans la base (pour masquer une présence sur une machine, par exemple). Pour ce faire, il faut donner le privilège TRIGGER à l'utilisateur QoQCoT_user sur la base QoQCoT, depuis la machine où tourne l'interface web. C'est ce que va faire ce script. Nous verrons plus bas comment créer effectivement le trigger.
- b) **INUTILE SI ON N'A JAMAIS EU DE V2.0, mais on va le faire quand même, pour avoir une belle structure de BD up-to-date** – Modifier les propriétés du champs idProcess de la table « Connexions » de façon à ce qu'il puisse être NULL. Ceci est nécessaire en mode hybride rsyslog/poussin-coq. En effet, en mode poussin-coq, l'info du numéro de process, qui servait en mode rsyslog à appairer les événements de connexion et déconnexion, n'a plus de sens, du

coup elle n'est pas renseignés lors des requêtes d'insertion dans la base par le coq.

Conséquence, surtout si votre serveur MySQL est en mode « strict » : il faut que `idProcess` puisse être NULL. Ainsi, les 2 modes pourront cohabiter.

- c) Supprimer la table inutile « Aliases ». Nous avons créé historiquement cette table en prévision, dans une version future, de la gestion le problème épineux du renommage des machines et, plus généralement, de l'évolution dans le temps des salles machines dont on mesure le taux d'utilisation. Mais notre réflexion a évolué et nous avons changé d'algorithme : la version V4.0 Rocky règle ce problème sans cette table qui ne sera finalement jamais utile.
- d) **INUTILE SI ON N'A JAMAIS EU DE V2.0** - La « raccourcissation » (n. m. : cœur de métier des raccourcissologues) : transformation des noms longs (FQDN) en noms courts (FQDN tronqué avant le premier « . ») dans les tables « Connexions » et « MachinesToSalles ». C'est crucial en cas d'upgrade v2 → v3 car en mode v2, ce sont les noms longs qui sont insérés dans la base tandis qu'à partir du mode poussin-coq, ce sont les noms courts.

Maintenant que nous savons POURQUOI il faut exécuter ce script, voyons COMMENT :

5.1 Exécution du script `ModifieDBV2toV3.sh`

5.1.1 copier `/tmp/qoq-`

`cot/ressources/UPGRADE_FROM_PREVIOUS_VERSIONS/ModifieDBV2toV3.sh` sur un serveur Linux disposant du client MySQL ET depuis lequel l'utilisateur `root` de MySQL ET l'utilisateur `QoQCoT_user` sont autorisés à se connecter (table `user` de la base `mysql`). Par exemple le serveur MySQL, ou le serveur qui tourne l'interface web QoQ-CoT...

5.1.2 copier dans le même répertoire votre fichier `config.php` (si vous avez suivi les indications, il est dans `/tmp/qoq-cot/src/`) : le script utilise ce fichier pour récupérer toutes les données utiles à la création de la base.

5.1.3 Désactivez le cron de `peuplade.sh` (résidu du fonctionnement anté-poussin/coq, il y a de fortes probabilités qu'il ne soit pas ou plus présent dans votre installation).

5.1.4 lancer `./ModifieDBV2toV3.sh` dans ce répertoire. À noter :

- Vous serez interactivement sollicité pour toutes les actions exécutées par le script. À chaque fois, vous aurez le choix d'effectuer ou non l'action.
- Le script peut être lancé un nombre quelconque de fois, les actions pouvant être rejouées indéfiniment sans dommage aucun pour les données.

Les actions, dans l'ordre d'apparition à l'écran :

5.1.4.1 : **NORMALEMENT DÉJÀ FAIT LORS DE L'INSTALL de la V3.0, donc dire NON au script sauf si vous aviez sauté cette étape à l'époque** - Ajout du droit `TRIGGER` : le script vous demandera d'indiquer sur quelle machine tourne l'interface web QoQ-CoT afin de pouvoir paramétrer dans MySQL le droit `TRIGGER` sur la base `QoQCoT` depuis ladite machine pour l'utilisateur `QoQCoT_user`. Une fois l'action terminée, `QoQCoT_user` pourra créer le trigger, ça va se passer en 5.2.

5.1.4.2 : Modification des propriétés du champs `idProcess` de la table « Connexions » : bien qu'inutile dans votre cas, faites-le, vous aurez une structure de base up-to-date, c'est toujours mieux.

5.1.4.3 : Suppression de la table inutile « Aliases »

5.1.4.4 : **INUTILE POUR VOUS, dites fermement NON au script.** - La « raccourcissation » : transformation des noms longs (FQDN) en noms courts (FQDN tronqué avant le premier « . ») dans les

tables « Connexions » et « MachinesToSalles »

5.1.5 C'est terminé. Vous pouvez éventuellement détruire les copies de `ModifieDBV2toV3.sh` et `config.php`, ils ont fini leur travail, furtif et crucial à la fois.

5.2 NORMALEMENT DÉJÀ FAIT LORS DE L'INSTALL de la V3.0 - Création du trigger : cette opération va se faire au travers du script `/tmp/qoq-cot/src/setup.php`. Jetez un œil sur [la doc d'installation](#) si vous avez besoin de vous rafraîchir la mémoire au sujet du format du fichier `.csv`.

5.2.1 Retrouver le fichier csv décrivant vos salles, ou mieux (depuis V3.0 Commandant Poulard), régénérez-le dynamiquement à partir des données en base, en lançant `/tmp/qoq-cot/src/export_salles_to_csv.php > /chemin/vers/mon/fichier.csv`. Profitez-en pour le modifier/compléter si besoin pour refléter l'évolution de votre structure, très dynamique.

5.2.2 Lancer `/tmp/qoq-cot/src/setup.php /chemin/vers/mon/fichier.csv`

5.2.3 Ceci va régénérer les salles dans la base en fonction des données du fichier `.csv` (pas de changement si le fichier `.csv` n'a pas changé...) ET créer le trigger

Et voilà, c'est déjà fini, le trigger est créé tout comme il faut, vos données sont désormais protégées et vous avez gardé les mains propres et les ongles nets. Merci qui ???

6. Modifiez votre base de données pour la rendre v4.0 compatible. À nouveau profondément désireux de contribuer à rendre votre vie plus simple et plus slow, nous avons regroupé et automatisé toutes ces modifs dans le seul script `/tmp/qoq-cot/ressources/UPGRADE_FROM_PREVIOUS_VERSIONS/ModifieDBV3.1toV4.sh`.

L'objectif est double :

- Surtout, modifier la table `MachinesToSalles` afin qu'elle intègre deux nouveaux champs : les dates d'apparition et de disparition de la machine dans la salle. Le stockage de ces dates permet d'implémenter LA nouveauté Rocky V4 : la gestion de l'évolution dans le temps des machines présentes dans les salles afin de fournir des statistiques toujours plus justes et intensifier encore l'orgasme de votre N+1 qui lorsqu'il génère un graphe...
- Accessoirement, corriger un tout petit oubli des versions 3.0 et 3.1 : le scripts de création de la DB QoQ-CoT créait un utilisateur MySQL avec tous les bons droits SAUF, oubli de notre part, le droit `DELETE`. Conséquence principale : impossible de détruire un admin de la base de données depuis l'interface web... On ajoute donc ici ce droit manquant.

6.1 Exécution du script `ModifieDBV3.1toV4.0.sh`

6.1.1 copier `/tmp/qoq-cot/ressources/UPGRADE_FROM_PREVIOUS_VERSIONS/ModifieDBV3.1toV4.0.sh` sur un serveur Linux disposant du client MySQL ET depuis lequel l'utilisateur `root` de MySQL ET l'utilisateur `QoQCoT_user` sont autorisés à se connecter (table `user` de la base `mysql`). Par exemple le serveur MySQL, ou le serveur qui tourne l'interface web QoQ-CoT...

6.1.2 copier dans le même répertoire votre fichier `config.php` (si vous avez suivi les indications, il est dans `/tmp/qoq-cot/src/`) : le script utilise ce fichier pour récupérer toutes les données utiles à la création de la base.

6.1.3 lancer `./ModifieDBV3.1toV4.0.sh` dans ce répertoire. À noter :

- Vous serez interactivement sollicité pour toutes les actions exécutées par le script. À chaque fois, vous aurez le choix d'effectuer ou non l'action.
- Le script peut être lancé un nombre quelconque de fois, les actions pouvant être rejouées indéfiniment sans dommage aucun pour les données.

Les actions, dans l'ordre d'apparition à l'écran :

6.1.3.1 : Modification de la structure de la table MachinesToSalles

6.1.3.2 : Ajout du droit DELETE sur la base QoQCoT pour l'utilisateur QoQCoT_user

6.1.4 C'est terminé. Vous pouvez éventuellement détruire les copies de `ModifieDBV3.1toV4.0.sh` et `config.php`, ils ont fini leur travail, tout à la fois furtif et colossal.

7. Copiez votre `qoq-cot/.htaccess` actuel vers `/tmp/qoq-cot/src/.htaccess`

8. Renommer votre répertoire `qoq-cot` actuel en `qoq-cot.old`

9. déplacez `/tmp/qoq-cot/src` dans votre emplacement de prod (c.-à-d. le répertoire qui contient `qoq-cot.old`) et renommez-le en `qoq-cot`.

10. Ça y est, votre poulette a retrouvé ses 20 ans et est toujours toute à vous...

11. Mais ATTENTION, même si votre cocotte est fonctionnelle et vous génère (plus vite qu'avant) de beaux graphes, ça n'est pas encore terminé... Pour bénéficier de la puissance légendaire du Rocky, vous devez maintenant exploiter la spécificité de la version 4.0 : la gestion des durées de présence des machines dans les salles et donc la possibilité de prise en compte des vols, ajouts, renouvellements et tout autre variation du nombre de machines dans le temps. Voir [cette entrée de faq](#) pour plus de détails.

Désormais, une date de début et une date de fin de présence dans la salle sont associées à chaque machine. Vous devez maintenant renseigner ces dates, c'est la partie un peu ingrate...

Lors de l'upgrade de la DB en V4 de l'étape 6, toutes les machines se sont vues attribuer une date de début en 1970 et une date de fin en 2300 (via les valeurs par défaut de la table MachinesToSalles pour ces 2 champs). Ceci permet d'avoir une DB exploitable sans erreur par les scripts 4.0 qui utilisent ces champs, mais ne correspond évidemment pas à la réalité des dates d'apparition/disparition des machines dans les salles...

Vous devez donc exporter vos salles et machines actuelles vers un fichier csv en lançant depuis le répertoire `qoq-cot` `export_salles_to_csv.php > /chemin/vers/mon/fichier.csv`, puis compléter/modifier le fichier pour qu'il reflète l'évolution de vos salles en fonction des dates réelles d'apparition/disparition des machines. Par exemple, si vous aviez seulement les nouvelles machines pour la salle X, apparus le 12/01/2018 lors du renouvellement des anciennes, il vous faut ajouter ces anciennes machines en indiquant leur date d'apparition dans la salle comme date de début et le 12/01/2018 (ou le 11/01/2018, si elle n'ont pas servi le jour du renouvellement) comme date de fin.

Il ne restera ensuite plus qu'à injecter cette réalité dans le Rocky 4 pour qu'il se déchaîne : lancer, toujours depuis le répertoire `qoq-cot` `setup.php /chemin/vers/mon/fichier.csv`

C'est terminé, vous avez libéré toute la puissance du Rocky qui trépignait : toutes les évolutions de vos salles seront désormais prises en compte dans les graphes générés, pour des stats toujours plus justes et pertinentes. Gare à l'abus d'orgasmes chez le DSI ou le DGS vieillissant, il peut en mourir.

Donc pas plus d'un ou deux graphes Rocky par jour. Nous déclinons toute responsabilité en cas de dépassement de cette posologie max...

And now, ladies and gentlemen, enjoy the legendary QoQ-CoT Rocky 4 power !! 😊

Cas A2. J'upgrade depuis V3.0 qui était un upgrade d'une V2.0 ou inférieure = j'ai des connexions collectées par une version < V3.0 dans ma BD

J'upgrade d'abord en V3.1 en suivant [cette superbe doc](#).

Je me retrouve alors dans le cas A1, je peux donc suivre [la procédure détaillée dans la section précédente](#) pour finalement upgrader en v4.0 Rocky.

Je l'aurais mérité cette poulette neuve toutes options... Elle en vaut la peine. Elle est belle. Elle est aimante. Elle est tout pour moi.

Cas A3. J'upgrade depuis V2.0 Bernadette

J'upgrade d'abord en V3.1 en suivant [cette superbe doc](#).

Je me retrouve alors dans le cas A1, je peux donc suivre [la procédure détaillée dans la section précédente](#) pour finalement upgrader en v4.0 Rocky.

Bon 2h de manip, j'y suis... Ça a intérêt à être bien... Sinon je descend à Marseille et le leur fais manger crue moi, leur poulet fada !!

Cas A4. J'upgrade depuis une version antérieure à V2.0 Bernadette

Vous devez d'abord [passer en 2.0 Bernadette](#) puis suivre le cas A3 ci-dessus... Oui c'est long. Mais en même temps, vous avez un peu attendu, non ?? Au moins 2 ans et demi... Sans doute par amour de Bernadette, nous pouvons comprendre. Mais maintenant, il faut la quitter. Pas le choix. Et s'installer avec Rocky. Ça va vous changer...

B. Le plus long, les poussins et le coq

Cas B1. J'upgrade depuis V3.0 Commandant Poulard ou V3.1 Commandant Poulard

Vous êtes déjà en mode poussin/coq, le plus dur est fait 😊

Il y a cependant quelques nouveautés (bio) dans les poussins et le coq V4.0.

Côté poussin

Nouveauté V4.0 Rocky

- il y a dans la conf. du poussin V4.0, soit le fichier `poussin.yml`, un nouveau paramètre : « `exclude` ». Il prend en argument une [expression régulière type PERL](#). Tous les connexions avec des logins/noms d'utilisateurs qui matchent cette expression régulière ne seront pas considérées. Par exemple
 - `exclude` : `^DWM`
- permettra de ne pas prendre en compte les connexions initiées par des utilisateurs dont le login commence par « `DWM` », ce qui sera bien utile sous W10 par exemple (`DWM` = Desktop Window Manager)...
- Notons que ce paramètre se retrouve aussi dans la conf. du `coq`, soit `coq.yml`. Il est tentant à première vue de le faire figurer plutôt au niveau `coq`, une seule fois. Mais, car il y a toujours un `mais`, on augmentera alors le trafic réseau entre les poussins et le `coq`, puisque des connexions non voulues seront remontées vers le `coq` avant de finalement être ignorées... À vous de voir si vous êtes munis d'un gros tuyau, ou seulement d'une petite durite...

Nouveauté depuis la V3.1, donc nouveau pour vous si vous venez de V3.0

- il y a dans la conf. du poussin depuis la V3.1 un nouveau paramètre : « `hostname` ». Il permet de répondre à un cas auparavant non géré : il peut arriver que l'on souhaite que la machine remonte dans la base avec un nom différent de celui connu par le système (`hostname` sous linux, Nom de l'ordinateur sous Windows...)
 - Un exemple de cas où ça peut être utile : machine en double boot avec un nom système différent sous les 2 OS. Si on ne remplit pas ce paramètre, la machine va remonter dans la base sous 2 noms différents selon qu'elle tourne sous 1 OS ou l'autre, ce qui peut être ennuyeux pour les statistiques d'utilisation... On règle le problème en remplissant ce paramètre, dans le `poussin.yml` de l'OS #2, avec le nom système de la machine sous l'OS #1 (sous l'OS #1, on laisse le paramètre à vide).
 - Ce paramètre permet aussi de répondre au cas inverse : différencier des machines aux mêmes noms courts, dans des domaines différents : `toto.chicken.org` et `toto.fried.chicken.org`. On peut par exemple, pour la 2de, renseigner le poussin avec
 - `hostname` : `totofried`
 - afin que les connexions sur cette machine apparaissent avec le nom de machine « `totofried` » dans QoQ-CoT.
- **ATTENTION** : si vous renseignez le paramètre `hostname`, alors ce nom **doit**, comme celui de toutes les autres machines, apparaître dans le fichier `.csv` donné en entrée à `setup.php`, sinon les connexions issues de cette machine n'apparaîtront jamais dans les graphes... Dans le cas du double boot vu ci-dessus, vous avez logiquement pensé à mettre le nom de machine dans le fichier `.csv`, puisqu'il s'agit du nom système fourni par l'un des deux OS. Mais si l'on souhaite, pour d'autres et obscures raisons, changer le nom de machine remonté par un nom *qui n'est le nom système d'aucune des machines du parc surveillé*, alors il ne faut pas oublier de faire figurer ce nom dans le fichier `.csv`...
- **ATTENTION 2** : il est possible de renseigner le paramètre `hostname` avec un nom long, ce qui est désormais un des 2 seuls cas permettant d'introduire un nom long dans la base QoQ-CoT¹⁾. C'est tout à fait possible, mais plutôt déconseillé dans un monde de noms courts... Mais si vous avez vos raisons, pourquoi pas 😊

Si vous avez besoin des fonctionnalités `hostname` ou `exclude` sur certaines de vos machines,

installez-y (selon la procédure décrite dans le README.TXT du répertoire ressources/poussin-coq/poussin) le poussin en V4.0 ainsi que le poussin.yml associé (le même qu'avant, plus les paramètres hostname ou exclude dûment remplis). Sinon, vous pouvez garder votre vieux poussin 3.0, dont le jaune est un peu moins brillant mais qui reste toujours bien côté à l'argus. De même que vous pouvez conserver le 3.1 si vous n'avez pas besoin du paramètre exclude...

Côté coq

- Il y a dans la conf. du coq Rocky V4.0, soit le fichier coq.yml, un nouveau paramètre : « exclude ». Il prend en argument une [expression régulière type PERL](#). Tous les connexions avec des logins/noms d'utilisateurs qui matchent cette expression régulière ne seront pas considérées. Par exemple
 - exclude : ^DWM
- permettra de ne pas prendre en compte les connexions initiées par des utilisateurs dont le login commence par DWM, ce qui sera bien utile sous W10 par exemple (DWM = Desktop Window Manager)...
- Notons que ce paramètre se retrouve aussi dans la conf. du poussin, soit poussin.yml. Il est tentant à première vue de le faire figurer plutôt au niveau coq, une seule fois. Mais, car il y a toujours un mais, on augmentera alors le trafic réseau entre les poussins et le coq, puisque des connexions non voulues seront remontées vers le coq avant de finalement être ignorées... Comme vu plus haut, c'est à vous de juger : vous êtes doté d'un gros tuyau, ou seulement d'une petite durite...

Depuis la V3.1, le coq est livré également en version compilée coq32 et coq64, pour éviter d'avoir à installer les modules PERL. Mais, nous avons constaté, sans vraie explication technique à vous fournir, que ces versions compilées ne s'exécutaient pas sous certains environnements... Si vous êtes dans le cas d'un tel contexte récalcitrant, utilisez à la place le fichier SOURCES/coq.pl qui lui, fonctionnera... sous réserve de disposer des modules PERL suivants : strict, warnings, IO::SocketEncode, DBI, YAML::Tiny, Data::Dumper, MIME::Base64 et integer. La plupart devraient être installés de base, sauf probablement YAML::Tiny.

Cas B2. J'upgrade depuis V2.0 Bernadette

Vous devez d'abord [passer en 3.1 Commandant Poulard](#)...

Cas B3. J'upgrade depuis une version antérieure à V2.0 Bernadette

Vous devez d'abord [passer en 2.0 Bernadette](#)... Vous sera alors dans le cas B2, qui vous permettra d'upgrader en V3.1 Commandant Poulard pour enfin passer en 4.0 via la procédure du cas B1. Long sera le chemin, mais pensez qu'il vous conduira, c'est tout à fait garanti, vers un bonheur sans nuage, peuplé de volatiles malins autant que bienveillants et de DSI extatiques.

1)

L'autre cas : une machine Linux - en mode poussin/coq ET dont le paramètre hostname de

`poussin.yml` n'est pas renseigné - est (mal) configurée avec un nom long dans `/etc/hostname` (qui doit être normalement renseigné avec un nom court). En effet, le module Perl utilisé dans le poussin Linux remonte comme *nom court* le contenu du fichier `/etc/hostname`.

From:

<https://sourcesup.renater.fr/wiki/qoq-cot/> - **QoQ-CoT : DoC**

Permanent link:

https://sourcesup.renater.fr/wiki/qoq-cot/upgrade_to_v4.0_rocky



Last update: **2018/06/22 01:21**