

PROLOGUE (important...)

À l'issue de votre install, merci de ne pas oublier de vous connecter sur [SourceSup](#) et de nous laisser un message ("Commencer une nouvelle discussion") [dans le forum « Poulailier »](#) du projet sur SourceSup, en indiquant l'établissement que votre QoQ-CoT va désormais couvrir, le nombre de postes concernés et toutes autres infos/cris/cocoricos de joie que vous jugerez utiles. D'une, vous apparaîtrez sur [la carte](#) (et ça, c'est la classe à Dallas ! 😊) et de deux, ça nous permettra de mieux connaître et de communiquer plus directement avec les utilisateurs de la solution (nouvelles features, évolutions, bugs...). Merci 😊

L'INSTALLATION : et la QoQ-CoT fut...

L'installation de la « solution » QoQ-CoT se décompose en 2 parties distinctes (que l'on cernerait bien mieux si l'on se fend d'un rapide coup d'œil à la [présentation générale](#)) :

1. l'interface web qui va permettre d'exploiter les données de connexion stockées dans la base MySQL et de produire ces graphes qui font rêver. Cette partie de l'installation est documentée ci-dessous.
2. la récupération des données de connexion sur les machines du parc surveillé, assurée, depuis la version 3.0 Commandant Poulard, par les poussins et le coq : ce sont les clients QoQ-CoT (poussins) pour les différents OS, à installer sur chaque poste surveillé, et le serveur (coq), que nous suggérons fortement d'installer sur le serveur MySQL, auquel les poussins remontent (trafic chiffré) les données de connexions récoltées sur le parc surveillé. Enfin, le coq - lui seul en a le droit - inscrit ces données dans la base MySQL. Pour l'installation du coq et des poussins, référez-vous au répertoire ressources/poussin-coq de l'archive.

Vous DEVEZ commencer par installer l'interface web, puis le coq et enfin les poussins.

Procédure d'installation de l'interface web de QoQ-CoT

Prérequis :

- un serveur LAMP correctement configuré
 - version de MySQL $\geq 5.1.73$
 - si PHP 5
 - version de PHP $\geq 5.4.45$
 - les modules PHP `php5-common`, `php5-cli`, `php5-mysql`, `php5-gd`, `php5-ldap` et `libapache2-mod-php5`.
 - si la version de PHP est $\geq 5.5.7$, ajouter le module `php5-xml`
 - si PHP 7.x
 - les modules PHP `php7.x-common`, `php7.x-cli`, `php7.x-mysql`, `php7.x-gd`, `php7.x-ldap`, `php7.x-mbstring`, `php7.x-mcrypt` (inutile si $x > 0$), `php7.x-xml` et `libapache2-mod-php7.x`
- **IMPORTANT 1** : assurez-vous que les erreurs PHP ne sont pas activées pour le répertoire dans lequel vous allez installer QoQ-CoT sur votre serveur web (ni par la configuration de votre serveur web, ni par un éventuel `.htaccess`). En effet, les éventuelles erreurs¹⁾ peuvent perturber la génération des graphes, modifiant l'image générée et la rendant impossible à

afficher. Pas mal de sollicitations de support de la part des QoQ-CoTiens sur ce problème 😊

- **IMPORTANT 2** : dans le fichier `php.ini` (chemin sous Debian/Ubuntu : pour PHP7 `/etc/php/7.x/apache2/php.ini` (pour PHP5 `/etc/php5.x/apache2/php.ini`), l'instruction `date.timezone` doit impérativement être définie, sous peine de graphes désespérément vides... Exemple :

```
[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = "Europe/Paris"
```

1. Copiez le répertoire `src` de l'archive où vous le souhaitez dans votre serveur web et renommez-le en `qoq-cot`.
2. Renommez le fichier `config.php.dist` en `config.php`.
3. Éditez le fichier `config.php` et renseignez les différentes constantes qui servent à paramétrer votre QOQ-COT rien qu'à vous.

Quelques explications pour le paramétrage de `config.php`, où sont définies diverses constantes :

- [Constante ADMIN] Il vous faut renseigner le login d'un usager qui bénéficiera des droits « admin » : accès aux données de connexions nominatives et... droits d'ajout/suppression des admins. C'est donc cet usager qui devra se connecter pour ajouter, grâce à l'onglet « ADMIN » de l'interface, tous les autres admins nécessaires.
- [Constante SITES] Ici, on va définir les « sites » de votre organisation. **Nouveauté** (parmi tant d'autres...) de la version 5.0 Mac Bec, le concept de « site » est relativement proche de son sens commun : si votre parc machines s'étend sur plusieurs *sites* géographiques distincts, vous pouvez refléter cette organisation dans QoQ-CoT. Intérêt : si vous avez 300 salles sur 6 sites distincts, possiblement gérées par des équipes différentes, il est sans doute plus pertinent de pouvoir restreindre sa consultation de QoQ-CoT à un site ou un autre. Mais en même temps, vous gérez tout ça depuis une seule instance de l'interface web QoQ-CoT depuis laquelle vous pouvez basculer à loisir d'un site à l'autre, ou même choisir de les voir tous ensemble. Bref, les avantages de la consolidation, tout en gardant la possibilité d'une vue locale : qui voudrait devoir choisir entre l'aile ou la cuisse quand on peut avoir les deux ?!! 😊
 - si vous n'avez qu'un site, il suffit de ne pas faire figurer la ligne définissant les SITES
 - si vous en avez plusieurs, vous les déclarez tous dans l'"array" comme ceci :

```
define('SITES', serialize(array('Site1', 'Site2', ..., 'SiteN')));
```
- [Constantes SQL_DBNAME, SQL_HOST, SQL_PORT, SQL_USERNAME, SQL_PASSWORD] Il s'agit, vous l'aurez compris, des informations nécessaires (nom de la base, du serveur MySQL associé et son port, utilisateur MySQL et password...) pour la connexion à la base de données où QoQ-CoT, plus précisément le `coq`, stocke les informations de connexion des utilisateurs, reçues par les poussins. Nous suggérons, sans grande originalité, d'appeler la base « QoQCoT » (*pas de tirets « - » autorisés dans les noms de bases MySQL...*) et l'utilisateur « QoQCoT_user » (c'est ainsi que nous y ferons référence dans ce qui suit). La création de QoQCoT et de QoQCoT_user se fera dans l'étape suivante.

Ces informations DOIVENT être les mêmes que celles spécifiées dans le fichier `coq.yml` de configuration du `coq` (voir le fichier `ressources/poussin-coq/coq/README.TXT`).
- [Constante CONNEXIONS_CACHE] **Nouveauté** de la version 5.0 Mac Bec. Nous avons constaté

que les performances pour l'affichage des graphes pouvaient chuter notablement avec l'augmentation massive du nombre de connexions présentes en base (table *Connexions*). Pour un grand site et quelques années, on peut facilement atteindre plusieurs million de connexions... Afin de diminuer le temps d'affichage, il est désormais possible de générer une table annexe en cache qui ne contient que les connexions des X derniers jours glissants, X étant fixé par la valeur de la constante `CONNEXIONS_CACHE`. C'est très intéressant si on sait qu'on travaille *principalement* sur une fenêtre temporelle correspondant par exemple au 3 derniers mois voire au dernier mois. Les requêtes se feront alors sur la table de cache, qui sera d'une taille extrêmement réduite par rapport à la table *Connexions* originale. Le temps d'affichage est alors *drastiquement* amélioré. Évidemment, si l'on requête sur une période antérieure au nombre de jours indiqués dans la constante `CONNEXIONS_CACHE`, la requête se fera sur la table complète.

- si la valeur de `CONNEXIONS_CACHE` est fixée à 0, la table annexe en cache n'est pas mise en place
 - si on doit modifier ce réglage, il faudra lancer la commande "php setup.php -c" afin de régénérer la table de cache afin qu'elle soit en accord le nouveau nombre de jours choisis.
- [Constantes `JOURS_OUVRES`, `HEURE_DEBUT` et `HEURE_FIN`] Pour configurer les valeurs par défaut concernant les jours ouvrés et la fenêtre horaire d'observation quotidienne afin de coller au plus près à votre réalité. Ces défauts pourront bien sûr être modifiés à loisir dans les formulaires des graphes avant de lancer le tracé.
 - [Constante `URL_ANNUAIRE`] Pour renseigner le motif général de l'URL des fiches annuaire des utilisateurs : vous devez indiquer l'URL d'accès standard à une fiche, en remplaçant le login de l'utilisateur par la chaîne `_LOGIN_`. Ce paramètre n'est pas obligatoire.

4. Créer la base de données `QoQCoT` et l'utilisateur `QoQCoT_user`. Ne pas oublier de donner les droits `INSERT`, `UPDATE`, `SELECT` et `TRIGGER` sur la base `QoQCoT` à l'utilisateur `QoQCoT_user` depuis la machine où est installé l'interface web `QoQ-CoT` ET depuis la machine où tournera le `coq` (nous suggérons fortement d'installer le `coq` sur le serveur MySQL). Pour vous éviter les traces de cambouis sous les ongles, vous pouvez, si vous le souhaitez, confier toute cette opération au script `ressources/INSTALL/CREATE_DB_AND_USER_QoQ-CoT.sh` de l'archive. Facile :

- **4.1.** copier `ressources/INSTALL/CREATE_DB_AND_USER_QoQ-CoT.sh` sur un serveur Linux disposant du client MySQL ET depuis lequel l'utilisateur `root` de MySQL est autorisé à se connecter (table `user` de la base `mysql`). Le serveur MySQL est un choix cohérent...
- **4.2.** copier dans le même répertoire le fichier `config.php` que vous avez soigneusement rempli : le script utilise ce fichier pour récupérer toutes les données utiles à la création de la base.
- **4.3.** lancer `CREATE_DB_AND_USER_QoQ-CoT.sh`
- **4.4.** le script vous demandera d'indiquer sur quelles machines tourneront 1) l'interface web `QoQ-CoT` et 2) le `coq` afin de pouvoir paramétrer dans MySQL les autorisations d'écriture sur la base `QoQCoT` depuis lesdites machines pour l'utilisateur `QoQCoT_user`.
- **4.5.** la base `QoQCoT` est désormais prête. Le `coq` va pouvoir la nourrir, l'interface web y puiser toutes les infos nécessaires pour générer tous ces merveilleux graphiques dont votre DSI est déjà fou.
- **4.6.** vous pouvez détruire les copies de `CREATE_DB_AND_USER_QoQ-CoT.sh` et `config.php`, rétrogradés en quelques secondes de héros providentiels à `has-been`.

5. Ajout des tables nécessaires dans la base, insertion de vos salles dans l'application et ajout du trigger sur la base qui va interdire toute mise à jour (forcément malicieuse, par exemple dans le but

de masquer une présence sur une machine) d'une date de fin de connexion par une date antérieure. Ces opérations, entre autres, sont effectuées grâce au script `qoq-cot/setup.php`.

Concernant l'insertion des salles, la source des données est un fichier au format CSV. Notons que depuis la version 2.0 Bernadette, ce script peut être relancé à loisir, sans que ni les données de connexion, ni les tables et le trigger déjà créés ne soient modifiés. Les opérations de création de tables n'ont lieu que si les tables n'existent pas encore et le trigger est reconstruit à l'identique. Seules les tables `Salles` et `MachineToSalles` sont vidées puis nourries avec les nouvelles données issues du fichier CSV : les salles sont ainsi redéfinies. Pratique pour ajouter/supprimer/modifier simplement des salles et groupes de salles.

Avec la notion de « **site** » introduite depuis la version 5.0 Mac Bec, les salles peuvent se gérer indépendamment par site géographique. Si l'on a décidé, dans son `config.php`, de segmenter son parc en sites (qui seront tous ensuite accessibles depuis une unique interface web QoQ-CoT, so magic !), alors il doit y avoir autant de fichiers CSV que de sites.

Les fichiers csv doivent être formatés de la façon suivante :

- Chaque ligne comporte 5 champs dans l'ordre suivant : “Nom de la machine”, “Nom de la salle à laquelle appartient la machine”, “Nom du groupe auquel appartient la salle”, “date de début de l'apparition de la machine dans la salle”, “date de disparition de la machine dans la salle”. Appelons « période de présence » l'intervalle de temps entre la date d'apparition et la date de disparition.
- Le séparateur est la **virgule** et les champs sont entre **double quotes**.
- Une machine peut **apparaître plusieurs fois** dans le fichier (dans la même salle ou dans des salles différentes) **si et seulement si les périodes de présence associées sont disjointes** (cas d'une machine qui a été retiré d'une salle pendant un certain laps de temps ou a été déménagée d'une salle à une autre avec un laps de temps de stockage entre les 2 périodes) **ou contiguës** (la date de fin d'une période est alors la date de début de l'autre : c'est le cas d'un déménagement le jour-même dans une autre salle, où la machine a pu être utilisée dans les 2 salles le même jour et doit donc apparaître dans les 2 salles ce jour-là...). **ATTENTION : cette contrainte d'unicité d'une machine pour une période de présence donnée n'est pas vérifiée automatiquement par QoQ-CoT. Il faut donc que VOUS fassiez en sorte de la respecter pour ne pas générer de situations incohérentes dans la base...**
- **“Nom de la machine” doit être un nom court** (c'est-à-dire toto si le nom DNS complet de la machine est `toto.domaine.org`) **et n'apparaître qu'une seule fois pour une période de présence donnée** en tant que nom machine dans le fichier. Si on a plusieurs machines au même nom court (`toto.chicken.org` et `toto.fried.chicken.org`), il faudra les différencier à l'aide du paramètre `hostname` de `poussin.yml` : par exemple, on indique `totofried` comme `hostname` sur le `poussin` installé sur `toto.fried.chicken.org` et on indique `totofried` comme nom machine dans la ligne correspondante du fichier csv. Une aide est fournie en commentaire dans le fichier `poussin.yml` qui vient avec l'archive, mais on pourra consulter pour plus de détails [cette section de la procédure d'upgrade](#). (Et si on s'intéresse à l'historique du nommage des machines dans QoQ-CoT, on aura beaucoup de plaisir à lire [cette entrée de FAQ](#)).

Si vous n'avez qu'**un seul site** (pas de définition de `SITES` dans votre `config.php`), lancer alors simplement le script `setup.php` : `setup.php /chemin/vers/votre/unique/fichier.csv`

Si en revanche vous avez défini **plusieurs sites**, lancer, pour chaque site, `setup.php` : `setup.php /chemin/vers/le/fichier.csv/du/site nom_du_site`, où `nom_du_site` doit être un des noms figurant dans la liste de sites `SITES` de votre `config.php`.

MAIS ATTENTION, ça n'est pas fini ! En effet, il faut savoir que depuis la version 5.0 Mac Bec, il y a une nouveauté importante au sein de `setup.php`. Comme vu plus haut, la version 5.0 Mac Bec introduit la possibilité de créer et d'utiliser une *table annexe de cache des connexions* pour optimiser le temps d'affichage des graphes. Si l'on donne une valeur X supérieure à 0 à la constante `CONNEXIONS_CACHE` dans `config.php`, on aura une table de cache et elle contiendra seulement les connexions des X derniers jours glissants (si X vaut 0, la table n'est pas créée).

Cette table est, comme les autres, créée, ou mise à jour si elle existait déjà, à l'aide de `setup.php` **MAIS uniquement** s'il est lancé avec l'option `-c`. Ainsi, pour que la table de cache soit créée et utilisée, **il vous faut, à l'installation de votre QoQ-CoT, lancer une 2de fois le setup (la 1re fois, c'était avec le fichier CSV en argument), cette fois avec l'option -c : `php setup.php -c`** (encore une fois, ceci n'a de sens que si vous avez défini une valeur supérieure à 0 pour `CONNEXIONS_CACHE` dans `config.php`). Notons que l'option `-c` de `setup.php` a pour effet de faire en sorte que seule les actions concernant la table de cache soient effectuées : le reste du code n'est pas exécuté²⁾.

Et de même, **chaque fois que l'on change la valeur de la constante `CONNEXIONS_CACHE`** (c'est-à-dire le nombre de jours X, sachant que la table de cache contient seulement les connexions des X derniers jours glissants) dans `config.php`, **il faut relancer `php setup.php -c`** pour que cette table soit mise à jour selon la nouvelle valeur.

6. Installation des autorisations d'accès

Créez, dans le répertoire `qoq-cot`, un fichier `.htaccess` qui va définir l'ensemble U des usagers qui pourront se connecter à votre cocotte : libre à vous d'utiliser tout mode d'authentification/autorisation. Astuce utile : faites en sorte que l'utilisateur choisi comme « admin » dans la config fasse partie de U, c'est mieux 😊

De nombreux exemples de fichiers `.htaccess` mettant en jeu divers mécanismes d'authentification/autorisation sont disponibles dans le répertoire `qoq-cot/ressources/EXEMPLES_`.`htaccess`

Vous le savez évidemment... mais rappelons quand même que pour que le fichier `.htaccess` soit pris en compte par Apache, il est nécessaire d'ajouter cette directive à la configuration de votre host Apache :

```
<Directory /votre/chemin/vers/qoq-cot>
    AllowOverride AuthConfig
</Directory>
```

À NOTER 2 petites faiblesses concernant le logout... :

1. si vous choisissez une authentification par CAS, le menu « Déconnexion » sera inopérant... Le seul moyen de se déconnecter consistera dans ce cas à fermer le navigateur ;
2. même punition si vous utilisez Internet Explorer (mais là, vous l'aurez bien cherché...), et cette fois quelle que soit votre méthode d'authentification...

7. Peuplement de la base

Une fois les tables `Connexions`, `Salles` et `MachinesToSalles` créées par le setup, il ne vous reste qu'à peupler la base.

Bonne nouvelle, vous n'avez rien à faire 😊 Les poussins et le coq vont travailler pour vous et remplir votre base au fil de l'eau sans action de votre part. Connectez-vous à l'interface sous le nom de l'utilisateur choisi comme « admin » dans la config et observez avec émotion les connexions naître, vivre et mourir...

Vous voici enfin prêt. Nous espérons que vous passerez de bons moments, studieux mais intimes, avec votre QoQ-CoT 😊

Procédure d'installation du coq et des poussins

Pour l'installation du coq et des poussins, référez-vous au répertoire `ressources/poussin-coq` de l'archive.

ÉPILOGUE (important...)

À l'issue de votre install, merci de ne pas oublier de vous connecter sur [SourceSup](#) et de nous laisser un message ("Commencer une nouvelle discussion") [dans le forum « Poulailier »](#) du projet sur SourceSup, en indiquant l'établissement que votre QoQ-CoT va désormais couvrir, le nombre de postes concernés et toutes autres infos/cris/cocoricos de joie que vous jugerez utiles. D'une, vous apparaîtrez sur [la carte](#) (et ça, c'est la classe à Dallas ! 😊) et de deux, ça nous permettra de mieux connaître et de communiquer plus directement avec les utilisateurs de la solution (nouvelles features, évolutions, bugs...). Merci 😊

1)

oui, oui, il ne devrait pas y en avoir, mais selon les différences de versions PHP et pchart, il peut en apparaître quelques-unes...

2)

Pour les curieux, voici ce que fait **exactement** la commande `setup.php -c` : **(1)** elle crée une table `Connexions_Cache` structurellement identique à `Connexions` avec les connexions des X derniers jours (oui, oui, *créée* à chaque fois... Et si elle existe déjà, elle est détruite puis recréée, ce qui a un intérêt de performance dans certains cas que nous n'aborderons pas ici) ; **(2)** elle crée un « trigger » pour que toutes les nouvelles connexions soient dupliquées dans la table `Connexions_Cache` ; **(3)** elle crée une *tâche planifiée* MySQL qui efface chaque jour les connexions > X jours dans la table `Connexions_Cache`.

From:

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

Permanent link:

https://sourcesup.renater.fr/wiki/qoq-cot/installation_v5.0_mac_bec



Last update: **2021/04/01 10:09**