

Un QFieldCloud auto-hébergé ?



Retour d'expérience sur l'installation et la configuration de QFieldCloud

René-Luc D'HONT, 3liz



QField & QFieldCloud



QField - Les projets QGIS dans la poche

L'application **QField** de la société **OpenGIS** permet de consulter des projets **QGIS** sur sa **tablette** ou son **smartphone**, et de faire des relevés sur le terrain via les formulaires d'édition.



QFieldCloud - Portail web pour synchroniser les projets et les données



QFieldCloud d'OpenGIS est la plate-forme de gestion centralisée des projets QGIS et des données collectées dans l'application **QField**

- Gestion d'équipe
- Envoi et partage de projets QGIS
- Synchronisation des données et des fichiers (photos, fichiers SIG)
- Consultation et synchronisation depuis QGIS, la plateforme ou QField



QField.Cloud ou QFieldCloud auto-hébergé ?

OpenGIS propose la mise à disposition de QFieldCloud à travers son instance officielle <https://qfield.cloud/pricing.html>

- Simplicité (maintenance, sauvegardes, etc.)
- Plusieurs tarifs en fonction des besoins.
- Un accès gratuit pour des usages simples

Pricing

QFieldCloud grows with your needs! Start with a free community plan, and upgrade when you need larger storage, more features or more access control on projects.

Monthly

EarlyBee*

Community

Free

Unlimited public/private projects
Basic data hosting and versioning
File based projects only

Pro

€15 / mo

Everything included in Community
Basic collaboration on private projects
Better data hosting and versioning
Offline PostGIS layers support

Organization

€20 / user / mo

Everything included in Pro
Unlimited collaboration
Team management
Minimum two users, pay only for additional active users
28 Days free trial





QFieldCloud auto-hébergé

QFieldCloud est **OPEN-SOURCE**, et peut être **installé sur un serveur** :

- **centraliser** les logiciels sur son serveur (Lizmap, PostgreSQL, etc.)
- **personnaliser** les configurations logicielles et matérielles
- **maîtriser** les coûts

Attention ⚠ Cela nécessite :

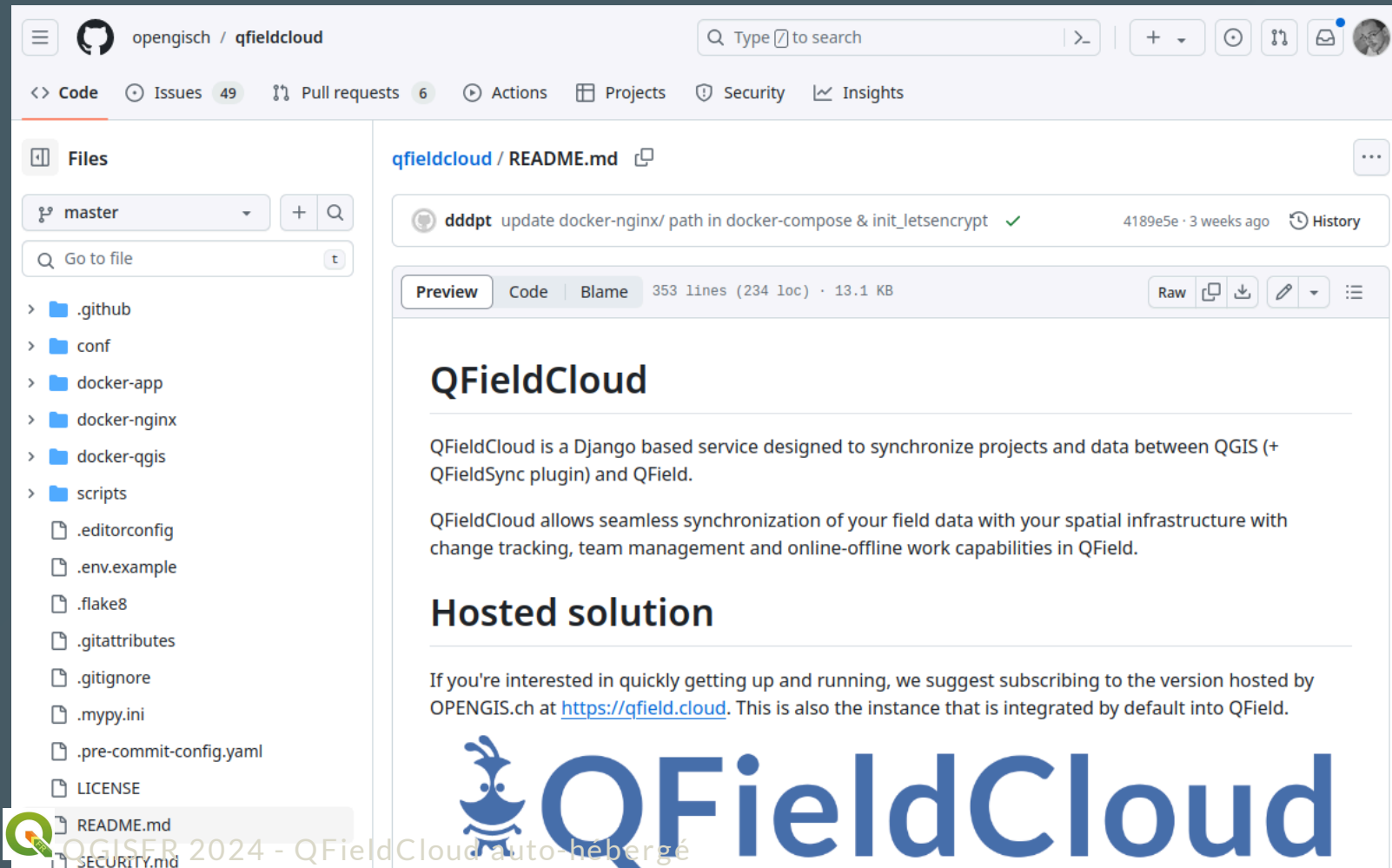
- des **compétences** en gestion d'infrastructure serveur et de déploiement de logiciels
- du **temps** pour la veille, la maintenance, la gestion
- une **responsabilité** sur la gestion des données, la disponibilité du service
- un **coût** pour la location du serveur, de serveurs de backup, de développement, etc.



Dépôt des sources

La procédure d'installation est accessible sur **Github** :

<https://github.com/opengisch/qfieldcloud/>



The screenshot displays the GitHub repository for **opengisch / qfieldcloud**. The main content area shows the **README.md** file, which describes QFieldCloud as a Django-based service designed to synchronize projects and data between QGIS (+ QFieldSync plugin) and QField. The README also mentions that QFieldCloud allows seamless synchronization of field data with spatial infrastructure, including change tracking, team management, and online-offline work capabilities. A section titled **Hosted solution** suggests subscribing to the version hosted by OPENGIS.ch at <https://qfield.cloud>, noting that this is the instance integrated by default into QField.

The repository structure is visible on the left, showing folders like **.github**, **conf**, **docker-app**, **docker-nginx**, **docker-qgis**, and **scripts**, along with various configuration files and a **LICENSE** file.

At the bottom of the image, there is a logo for **QFieldCloud** and a text overlay: **QGISER 2024 - QFieldCloud auto-hébergé**.

Pré-requis

L'installation est conseillée sur une machine **Linux** qui doit être **correctement administrée** : mises-à-jour, sécurité des accès, sauvegardes, etc.

Elle nécessite de **configurer le réseau** pour permettre l'accès depuis l'extérieur. Par exemple via <https://qfieldcloud.votredomaine.fr>

Plusieurs composants sont nécessaires pour faire tourner **QFieldCloud**

- L'outil **Docker** qu'il faut bien maîtriser
- Une **base de données PostgreSQL** pour stocker les configurations et les données de l'application : comptes, logs, projets, etc.
- Un **service de stockage de fichiers** (pour les projets QGIS, les données, les photos, etc.) compatible S3 (minio)
- l'**application Web QFieldCloud** (Django) et les services liés (nginx, memcached, etc.),
- le **service de synchronisation** qui tourne en tâche de fond



Docker

“ Docker est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels lancée en 2013. (Wikipédia, 2024) ”



Docker facilite l'installation de composants logiciels grâce à la conteneurisation. Un fichier **Dockerfile** permet :

- de lister les **applications** à lancer et leur "image" (qui "contient" l'application)
- de configurer les **variables** (URL, environnement, versions, répertoires, etc.)
- de placer les applications dans un **réseau dédié** pour les faire dialoguer
- On peut l'utiliser pour **démarrer** ou **stopper** automatiquement tous les composants.
- Attention à bien séparer les données dans des **volumes dédiés** (persistance)





Installation

Suivre le **README**

- **Installer Docker** sur le serveur
- Télécharger le **code source**
- Adapter le fichier d'**environnement**
- Construire les **images docker** et lancer les applications
- Installer la structure de la **base de données**
- Installer les **fichiers web** (CSS, JavaScript)
- Vérifier **le statut**
- Créer l'**utilisateur admin**

Un tutoriel récent en français :
<https://geotribu.fr/tags/#tag:qfieldcloud>



Exemple de commandes

- Récupérer les sources:

```
git clone --recurse-submodules git@github.com:opengisch/qfieldcloud.git
```

- Construire les images et lancer:

```
docker compose up -d --build
```

- Installer/mettre à jour la structure de la base

```
docker compose exec app python manage.py migrate
```

- Fichiers statiques (CSS, JavaScript)

```
docker compose run app python manage.py collectstatic --noinput
```

- Créer un utilisateur admin

```
docker compose run app python manage.py createsuperuser --username gis_admin --email gis_admin@votredomaine.fr
```

♥ Il faut aimer la ligne de commande ♥

Nous conseillons d'utiliser un fichier `Makefile` pour lancer plus facilement ces commandes, comme `make build` ou `make run`

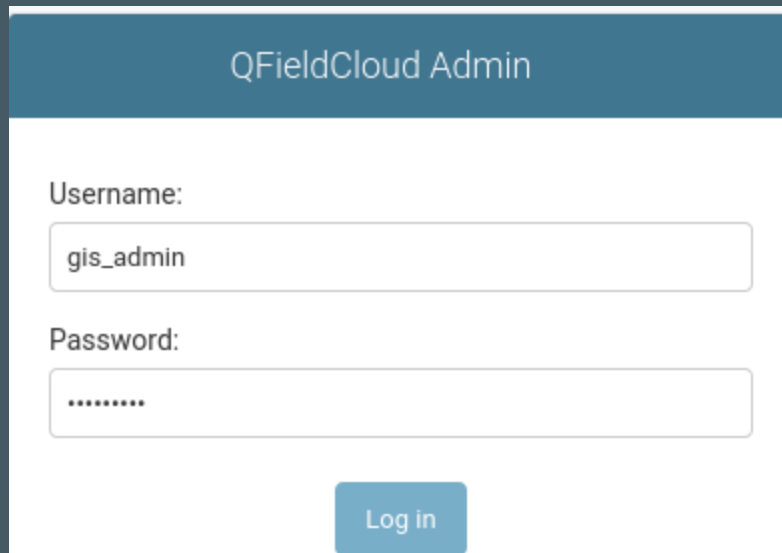


Créer le super-utilisateur

Pour créer un super-utilisateur, par exemple `gis_admin`, on lance :

```
docker compose run app python manage.py createsuperuser --username gis_admin --email gis_admin@votredomaine.fr
```

On peut ensuite se connecter sur l'interface d'administration accessible par défaut sur le port 7080 : <https://qfieldcloud.votredomaine.fr:7080>



The screenshot shows the 'QFieldCloud Admin' login page. It has a dark blue header with the text 'QFieldCloud Admin'. Below the header, there are two input fields: 'Username:' with 'gis_admin' entered, and 'Password:' with '*****' entered. A blue 'Log in' button is positioned below the password field.

À partir de là, vous pouvez accéder à tous les **menus de gestion**



Créer un utilisateur

On peut maintenant utiliser l'interface de QFieldCloud pour :

- créer des **groupes d'utilisateurs**
- créer un ou plusieurs **utilisateurs**
- créer des **organisations** et des **équipes**

✓ The person "mario" was added successfully. You may edit it again below.

Change person

mario

Storage: total: 10.0 GB; used: 0 bytes (0.00%); free: 10.0 GB

Username:

mario

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Password:

pbkdf2_sha256\$260000\$P1HnoSbiZuBHRX3

Email address:

mario@votredomaine.fr

First name:

Mario

Last name:

Bross



Souscriptions

Par défaut, **les nouveaux utilisateurs** ne sont pas activés tant que le statut de leur **souscription** n'est pas **Active payée** :

Change subscription

Subscription #3 user:mario plan:community total:10000MB

Plan: organization (organization) ✎ + ✖

Account: mario (Mario Bross) (UserAccount) ✎ +

Status: Active Paid ▼

Active since: Date: 2024-03-26 Today

On peut aussi modifier les différentes variables pour les **Plans** :

- taille maximum autorisée pour les fichiers,
- accès à une base de données spatiale,
- statut par défaut des nouveaux utilisateurs



Se connecter depuis QGIS

Installer l'extension
QFieldSync

Lancer le plugin
QFieldSync et configurer
l'accès :

double-cliquer sur
l'abeille

pour faire apparaître le
champ pour renseigner
son serveur


Se connecter: la liste des
projets apparaît

<https://docs.qfield.org/>



QGISFR 2024 - QFieldCloud auto-

QFieldCloud Sign In



The easiest way to transfer your project from QGIS to your devices! [Learn more about QFieldCloud.](#)


Server URL


Username

Password

☐ Stay signed in

New user? [Register an account.](#)

 Annuler

 Sign In

Les étapes de publication : créer le projet

- Créer un nouveau **projet QGIS** ou ouvrir un projet existant
- Configurer le **projet**
- Dans QFieldSync, **créer un projet** dans un répertoire local

QFieldCloud Projects Overview

Greetings [qfieldadmin](#).

You don't have any projects, create some by clicking the button in the bottom bar.

Choose how to create a new project

☐ Convert currently open project to cloud project (recommended)

A new QFieldCloud-compatible project is created from the currently opened QGIS project. In order to do so, datasets will be copied into an export directory that will act as your local mirror. Vector datasets will be converted to geopackage format to facilitate data synchronization from multiple devices while other dataset types will be copied to the new project location.

☒ **Create a new empty QFieldCloud project**

A new blank QFieldCloud project will be created. You will be responsible to move all the project-related files within the selected local directory, with the project file at its root. Project files will only be uploaded when you click the synchronize button. Make sure the selected directory contains no more than one QGIS project file.

Project Details

Name

Description

Owner

Local Project Settings

Local Directory

Les étapes de publication : synchroniser

- Ajouter des couches **vecteur** (fichiers ou PostgreSQL), **raster**
- Configurer les options des **couches** (formulaire, symbologie)
- Dans QFieldSync, **synchroniser vers le serveur**

QFieldCloud project name [qfieldadmin/test](#)

Local project directory

All files in the local project directory will be uploaded to QFieldCloud.
Now you should choose what actions to perform on those files by clicking on the checkboxes in the file list below.

Checking the "local" action (under the computer icon) or "cloud" action (under the cloud icon) will mark that file as single source of truth and will cause the file to be overwritten either in the cloud or the local storage. All other files will not be affected.

Filename			Action
projet_test.qgs	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Create file on the cl...
projet_test_attachments.zip	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Create file on the cl...

Prefer No Action Prefer Local Prefer Cloud

Aide Upload Files Annuler



QField

← QFieldCloud Projects



URL du serveur (Laissez vide pour utiliser le serveur par défaut)

http://localhost:7080/

Nom d'utilisateur ou e-mail

qfieldadmin

Mot de passe

.....

S'identifier

Nouvel utilisateur? [Enregistrer un compte.](#)

La manière la plus aisée de transférer votre projet de QGIS vers vos périphériques ! [En apprendre davantage sur QFieldCloud.](#)

Se connecter avec QField

On peut préciser le **serveur** à utiliser, puis les **informations de connexion**


QField propose ensuite de télécharger le projet

← QFieldCloud Projects

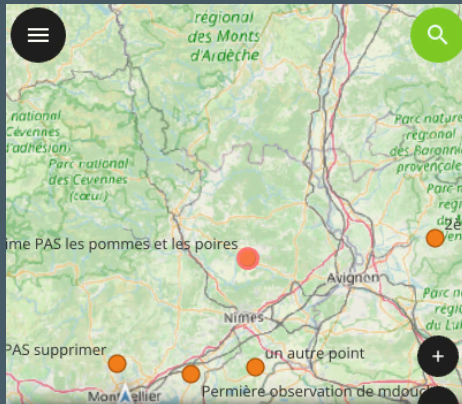
Bienvenue **qfieldadmin**.

Mes projets Communauté

qfieldadmin

 **test**
(QFieldCloud prépare les dernières données juste pour vous. Cela peut prendre un certain temps, veuillez patienter...)

Download project test



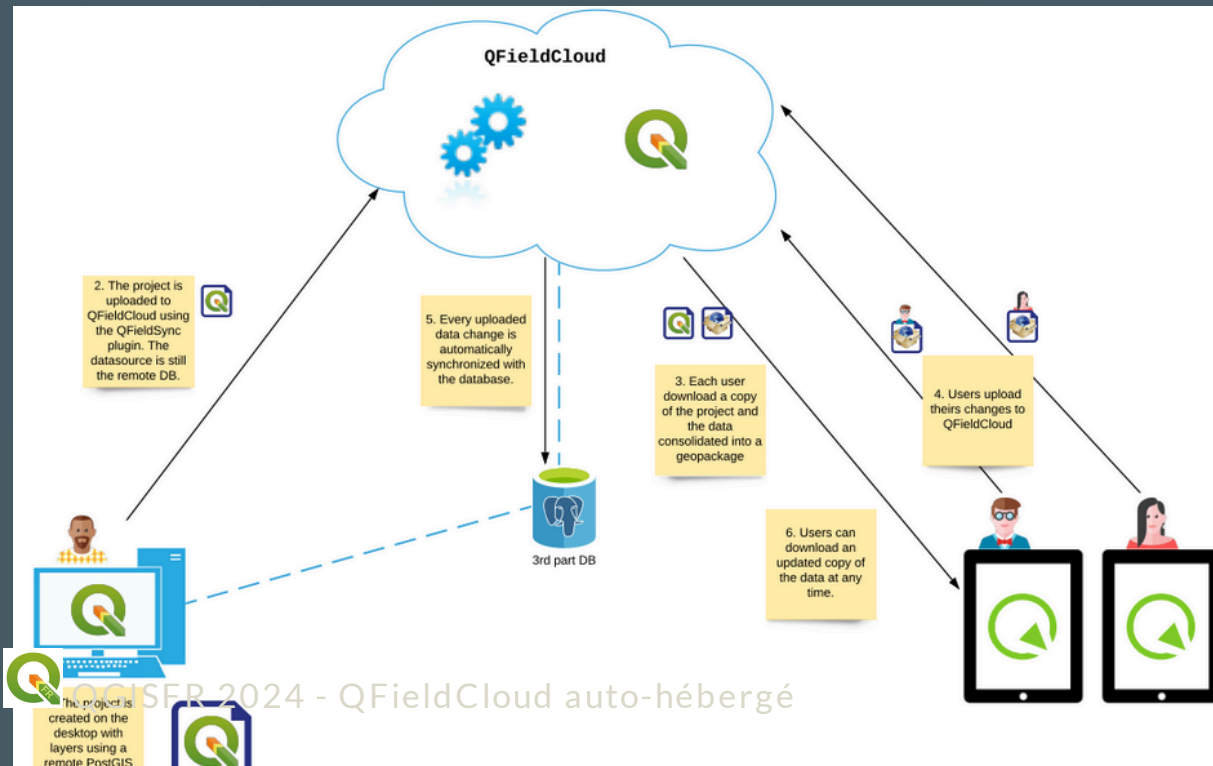
✓ TADA ✕

Id	3	⋮
titre	J'aime les pommes et les poires	⋮
date_observation	Tue Sep 20 02:00:00 2022 GMT+0200	⋮
observateurs	dupont	⋮

Synchroniser les données PostgreSQL

Si on utilise des couches **PostgreSQL** hébergées sur un serveur :

- on peut permettre aux utilisateurs **sur le terrain** de travailler **hors-connexion**,
- et **QFieldCloud** va pouvoir synchroniser leurs saisies **avec la base centrale** une fois qu'ils auront synchronisé leurs changements (connexion internet retrouvée)
- nous conseillons d'utiliser les **services PostgreSQL** et pas de connexion via login/mot de passe



Utiliser la synchronisation PostgreSQL

- Créer **nouveau projet QFieldCloud vide** dans un répertoire sur son poste de travail
- **Copier** le projet et les données existantes dans le **répertoire local** de ce projet vide
- Configurer les **couches PostgreSQL** qu'on souhaite éditer en mode
Édition hors connexion
- **Pousser vers QFieldCloud** avec le plugin QFieldSync
- Alors QFieldCloud pourra utiliser les informations des couches PostgreSQL stockées dans le projet QGIS pour faire opérer **la magie de la synchronisation**



Conclusion

Il est possible d'installer et d'utiliser **QFieldCloud** sur son propre serveur

Cela demande des **compétences** et du **temps**

Merci à **OpenGIS** pour la conception et la publication de ces outils !



Liens :

- Documentation : <https://docs.qfield.org/>
- Dépôt Github : <https://github.com/opengisch/qfieldcloud/>
- Discussions : <https://github.com/opengisch/QField/discussions/>
- 3liz: <https://3liz.com>



Merci de votre attention !

Vos questions sont les bienvenues !

