

Déploiement du Dashboard ECOLOGGING sur Internet (Render)

1. Objectif :

L'objectif est de rendre accessible sur Internet une plateforme web permettant :

- la récupération des données satellite CLS
- le stockage des mesures environnementales
- l'affichage en temps réel
- la visualisation graphique des données
- l'accès public via navigateur web

Cette plateforme constitue le centre de supervision du système **ECOLOGGING Satellite Station**.

2. Choix de la plateforme d'hébergement :

La plateforme **Render** a été choisie car elle permet :

- Hébergement gratuit
- Déploiement rapide
- Support Python/Flask
- HTTPS sécurisé
- Accès public mondial

Lien :

<https://render.com>

3. Crédit du compte Render :

1. Accéder au site Render
2. Cliquer sur *Get started*
3. Se connecter avec Google ou GitHub

4. Accéder au tableau de bord Render :

The screenshot shows the Render dashboard at dashboard.render.com. The user is in the 'Projects' section, with a profile icon for 'Mohamed's works...' and a search bar. The main area is titled 'Create a new Service' with three steps: 1 Choose service, 2 Configure, 3 Deploy. A 'Skip' button is available. Below the steps, there's a link 'Which to use?'. The page lists several service types with descriptions and creation links:

- Static Sites**: Static content served over a global CDN. Ideal for frontend, blogs, and content sites. Link: [New Static Site →](#)
- Web Services**: Dynamic web app. Ideal for full-stack apps, API servers, and mobile backends. Link: [New Web Service →](#)
- Private Services**: Web app hosted on a private network, accessible only from your other Render services. Link: [New Private Service →](#)
- Background Workers**: Long-lived services that process async tasks, usually from a job queue. Link: [New Worker →](#)
- Cron Jobs**: Short-lived tasks that run on a periodic schedule. Link: [New Cron Job →](#)
- Postgres**: Relational data storage. Supports point-in-time recovery, read replicas, and high availability. Link: [New Postgres →](#)
- Key Value**: Managed Redis®-compatible storage. Ideal for use as a shared cache, message broker, or job queue. Link: [New Key Value Instance →](#)

4. Crédit du dépôt GitHub :

Afin de déployer le projet, un dépôt GitHub est nécessaire.

Étapes :

1. Aller sur : <https://github.com/new>

2. Nom du dépôt :

```
ecologging-dashboard
```

3. Choisir Public

4. Cliquer sur **Create repository**

The screenshot shows the GitHub profile page for the user 'azallalmohamed'. The profile picture is a circular checkerboard pattern. The 'Repositories' tab is selected, showing two public repositories: 'ECOLOGGING' (updated 20 hours ago) and 'ecologging-dashboard' (Python, updated 5 days ago). The GitHub interface includes a search bar, typeahead dropdowns for 'Type', 'Language', and 'Sort', and a green 'New' button.

5. Ajout des fichiers du projet :

Fichier principal : app.py

Contient le serveur web Flask.

The screenshot shows the GitHub code editor for the 'app.py' file in the 'ecologging-dashboard' repository. The file contains 367 lines of Python code using the Flask framework. A sidebar on the right displays symbols and their definitions, such as 'DEVICE', 'AUTH_URL', 'API_URL', 'CLS_TOKEN', 'LAST_FETCH', 'app', 'db', 'conn', 'cur', and 'decode'. The commit history shows a recent update from '7c8bd7e · 5 days ago' by 'azallalmohamed'.

```

1 import os
2 import requests
3 import sqlite3
4 import time
5 import threading
6 import traceback
7 import pandas as pd
8 import plotly.graph_objects as go
9 from plotly.subplots import make_subplots
10 from flask import Flask, request, redirect, session
11
12 # -----
13 DEVICE = "278910"
14
15 AUTH_URL = "https://account.groupcls.com/auth/realm/cls/protocol/openid-connect/token"
16 API_URL = "https://api.groupcls.com/telemetry/api/v1/retrieve-bulk"
17
18 CLS_TOKEN = None
19 LAST_FETCH = None
20
21 app = Flask(__name__)

```

Fichier requirements.txt

Contient les bibliothèques Python nécessaires :

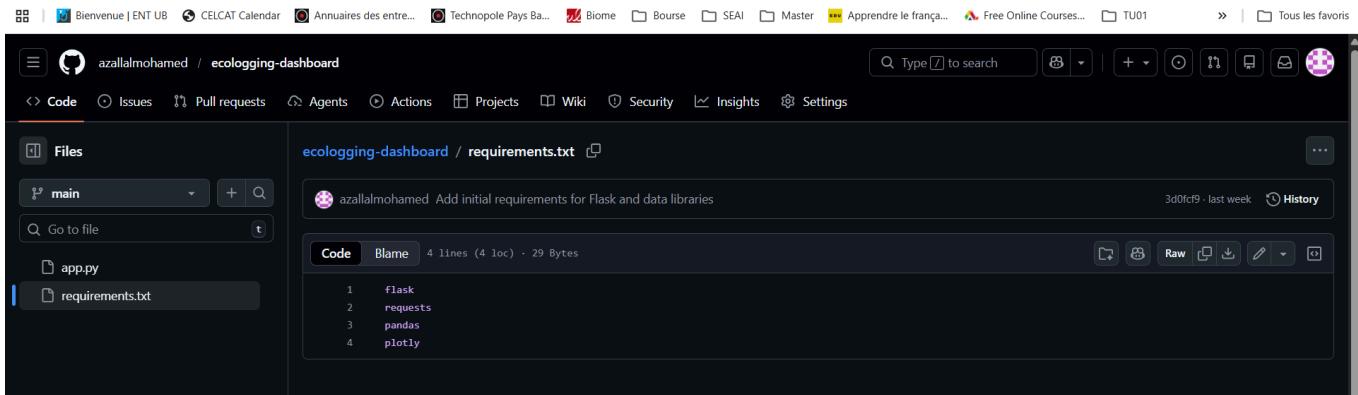
```

flask
requests

```

pandas
plotly

Ces fichiers sont envoyés sur GitHub.



The screenshot shows a GitHub repository named "ecologging-dashboard". The "Files" sidebar on the left lists "main", "app.py", and "requirements.txt". The "requirements.txt" file is selected and shown in the main pane. The content of the file is:

```
flask
requests
pandas
plotly
```

6. Crédation du Web Service sur Render :

Dans le tableau de bord Render :

1. Cliquer sur **New +**
2. Choisir **Web Service**
3. Connecter GitHub
4. Sélectionner :

ecologging-dashboard

7. Configuration du service :

Paramètres utilisés :

Paramètre	Valeur
Name	ecologging
Environment	Python
Build command	pip install -r requirements.txt
Start command	python app.py

Mohamed's works... My project / Production / ecologging + New ⚡ Upgrade ⚡ A

Settings

General

Name ecologging A unique name for your Web Service. [Edit](#)

Region Oregon (US West) Your services in the same region can communicate over a private network.

Instance Type Free 0.1 CPU 512 MB [Update](#)

Please enter your payment information to select an instance type with higher limits.

See remaining free usage or learn about free service limits.

General Build & Deploy Custom Domains PR Previews Edge Caching Notifications Health Checks Maintenance Mode Delete or suspend

Bienvenue | ENT UB CELCAT Calendar Annuaires des entre... Technopole Pays Ba... Biome Bourse SEAI Master Apprendre le frança... Free Online Courses... TU01 Tous les favoris

Mohamed's works... My project / Production / ecologging + New ⚡ Upgrade ⚡ A

Build & Deploy

Repository https://github.com/azallalmohamed/ecologging-dashboard [Edit](#)

Branch main [Edit](#)

Git Credentials azallalmohamed001@gmail.com (you) [Use My Credentials](#)

Root Directory Optional If set, Render runs commands from this directory instead of the repository root. Additionally, code changes outside of this directory do not trigger an auto-deploy. Most

General Build & Deploy Custom Domains PR Previews Edge Caching Notifications Health Checks Maintenance Mode Delete or suspend

Mohamed's works... My project / Production / ecologging + New ⚡ Upgrade ⚡ A

Build Filters

Include or ignore specific paths in your repo when determining whether to trigger an auto-deploy. Paths are relative to your repo's root directory. Learn more.

Included Paths Changes that match these paths will trigger a new build. [Edit](#)

+ Add Included Path

Ignored Paths Changes that match these paths will not trigger a new build. [Edit](#)

+ Add Ignored Path

Build Command \$ pip install -r requirements.txt [Edit](#)

Pre-Deploy Command Optional Render runs this command before the start command. Useful for database migrations and static asset uploads. [Edit](#)

\$

General Build & Deploy Custom Domains PR Previews Edge Caching Notifications Health Checks Maintenance Mode Delete or suspend

The screenshot shows the Render dashboard interface. On the left, there's a sidebar with navigation links such as Environment, Events, Settings (which is currently selected), MONITOR (Logs, Metrics), MANAGE (Environment, Shell, Scaling, Previews), Changelog, Invite a friend, Contact support, and Render Status. The main content area is titled 'ecologging' and contains sections for 'Start Command' (set to '\$ python app.py'), 'Auto-Deploy' (set to 'On Commit'), 'Deploy Hook' (a placeholder URL), and 'Custom Domains' (empty). On the right, there's a vertical sidebar with links for General, Build & Deploy, Custom Domains, PR Previews, Edge Caching, Notifications, Health Checks, Maintenance Mode, and Delete or suspend.

Puis cliquer sur :

Create Web Service

Render déploie automatiquement le serveur.

8. Mise en ligne :

Après déploiement, Render génère une URL publique :

<https://ecologging.onrender.com>

Cette URL permet d'accéder au dashboard depuis n'importe quel navigateur.

9. Fonctionnement global :

Le système complet fonctionne comme suit :

Capteurs → Arduino → Satellite → CLS
CLS → API → Serveur Render → Base de données
Base de données → Dashboard Web public

10. Résultat :

The screenshot shows the Render dashboard interface. On the left, a sidebar menu includes sections for Environment, ecologging (selected), Events, Settings, MONITOR (Logs, Metrics), MANAGE (Environment, Shell, Scaling), Previews, Changelog, Invite a friend, Contact support, and Render Status. The main area displays a log viewer for the 'ecologging' project. The log shows activity from February 9, 2026, at 2:04 AM, including messages like 'Your service is live', 'Available at your primary URL https://ecologging.onrender.com', and various HTTP requests. A search bar and filter options are visible at the top of the log viewer.

⚠ Pour se connecter, vous devez disposer d'un compte CLS avec accès au service Telemetry (contacter : useroffice@groupcls.zohodesk.eu).

Utilisez le même email et mot de passe que pour la plateforme officielle **CLS Environmental Monitoring**.

The screenshot shows the login page for 'ECOLOGGING - INRAe'. The title 'Connexion satellite CLS' is displayed above two input fields: 'CLS Email' and 'CLS Password'. A blue 'Connexion' button is located below the password field. The background is dark, and the text is white or light-colored.

 ECOLOGGING Station - INRAE

14.74 °C

5 %

998.6 hPa

840.0

ECOLOGGING Station – ID: 278910

