



Cloud computing

Travaux dirigés

Travailler mieux avec les SaaS

Utilisons codespace → <https://github.com/features/codespaces>

Connectez-vous à codespace et cloné un de vos dépôt GitHub (au hasard le TP CI/CD).

Faire et envoyer une modification pour ajouter un badge “Projet terminé ! 🚀” au README.md principal du dépôt.

```
### Rappel du module précédent ###  
  
# Cloner un dépôt  
git clone [ adresse de votre dépôt ]  
  
# Ajouter un changement à envoyer  
git add [ fichier dans le dépôt ]  
  
# Créer un commit avec ses changements  
git commit -m "mon petit message"  
  
# Envoyer les changements sur la  
# branche courante  
git push
```

Une calculette dans le Cloud ?

Version GitHub de ce sujet 📖

https://github.com/JeromeMSD/module_cloud-computing/blob/main/TD.md

À partir des échanges du CM2, nous allons réaliser une calculatrice prévue pour le Cloud. Pour cela nous passerons par trois étapes :

1. Création d'une API simple
2. Mise en place d'un serveur Redis.
3. Création d'une file d'attente pour ordonner les calculs.
4. Création d'un **Frontend** (pour les plus rapides d'entre vous).

On n'oublie pas les bonnes habitudes

1. Créer un dépôt privé avec un nom commençant par **4A_[ILC/SQR]_**
2. Ajoutez le second membre du binôme en tant que collaborateur.
3. Ajoutez-moi en collaborateur → jerome.massard@kiowy.com

Ni les bonnes pratiques

- README.md → pour votre vous du futur.
- un fichier **.gitignore** contenant ****/_pycache__**
→ pour ne pas embarquer le superflu.

Comme un goût de CI/CD



Concevoir une API simple (Python/Flask ou Rust)

Vous pouvez reprendre le projet du module précédent en base.

La calculatrice doit permettre les **additions**, les **soustractions**, les **multiplications** et les **divisions**.

Pour demander ces calculs :

- **POST** : Envoyer **un tuple** pour demander un calcul (*renvoie un id*)
- **GET** : Récupérer le résultat via un id.

Pour l'instant, les résultats sont stockés dans une variable dictionnaire.

L'API peut tomber 🤯

On ne va pas tout perdre pour un calcul impossible !

Utiliser Redis comme serveur de données, c'est un système de stockage clé/valeur. Il utilise le port **6379** pour communiquer.

Vous pouvez le lancer dans un conteneur via la commande

```
docker run --name myredis -p 6379:6379 redis
```

(utilisez un autre terminal)

Utilisez redis-cli pour accéder à redis et testez les commandes set, get, ...

L'API peut tomber 🤯

Connectons maintenant l'API à la Redis → [python et redis](#)

```
import redis
r = redis.Redis(host='localhost', port=6379, db=0)

# Ajout de variable
r.set('foo', 'bar') # Retourne True si réussite

# Lecture de
r.get('foo')        # Retourne la valeur de foo
```

Remplacer le stockage précédemment en variable par un stockage via Redis

Vous êtes déjà là ?! 🤯

#

Facultatif

Tout d'abord **Félicitation** ! Vous avez fini le sujet principal.

Ça vous dit un peu de design et d'implémentation d'interface ?

Une seule consigne :

Créer un frontend qui permet à l'utilisateur de saisir et d'envoyer des demandes de calcul à l'API.

Tips : vous pouvez utiliser le bon vieux combo **HTML/CSS/JS** ou ChatGPT
(pour gagner du temps uniquement 🙌)