


# Projet final

## Architecture Cloud et Serverless

 **Durée** : 4h00

 **Travail en binôme**

 **Objectif** : Développer une application avec une architecture serverless, tout en intégrant des services cloud.

Vous pouvez choisir l'un des **deux projets** proposés ou celui de votre choix.

 **Contraintes techniques** :

- ✓ Utilisation d'un **framework serverless** comme **Serverless, AWS SAM, SST**
- ✓ Intégration d'au moins **un service cloud** (ex: **S3, DynamoDB**)
- ✓ **Développement en local** avec une possibilité de déploiement cloud



## Projet 1 : Générateur de Memes



### Description

Développez une application permettant de créer des memes personnalisés à partir d'images et de textes fournis par l'utilisateur. L'application doit fonctionner avec une architecture **serverless** et générer un fichier image contenant le meme finalisé.



### Fonctionnalités attendues

- ✓ **Upload d'une image** (format JPG, PNG)
- ✓ **Ajout de texte** en haut et en bas de l'image
- ✓ **Génération d'un fichier final** contenant le meme personnalisé
- ✓ **Téléchargement du meme généré**

Service	Utilisation
<b>Object-Storage S3</b>	Stockage des images uploadées
<b>DBaaS</b>	Sauvegarde des memes générés (nom, texte, URL)

## **Projet 2 : Voicefaker - Générateur de Voix IA**

### **Description**

Développez une application qui permet de générer une phrase en audio avec une voix modifiée (ex: Dark Vador, robot, cartoon). L'application doit prendre une phrase en entrée et fournir un fichier audio transformé en sortie.

### **Fonctionnalités attendues**

- ✓ **Saisie d'un texte** en entrée
- ✓ **Génération d'un fichier audio** avec une voix modifiée
- ✓ **Téléchargement du fichier audio final**

Service	Utilisation
<b>Object-Storage S3</b>	Stockage des fichiers audio générés
<b>DBaaS</b>	Sauvegarde des fichiers générés (nom, texte, URL)
<b>FaaS</b>	Génération et modification de la voix

### **Consignes générales**

- ◆ Vous devez versionner votre code sur **GitHub**.
- ◆ Vous devez rédiger un **README** expliquant comment tester votre application.
- 💡 **Conseil** : Commencez par un **MVP (Minimum Viable Product)** avant d'ajouter des fonctionnalités avancées.

### **Ce que vous devez rendre**

 **Code source** sur un dépôt GitHub/GitLab

 **README.md** avec :

- ✓ Explication du projet
- ✓ Instructions d'installation et d'exécution

✓ Exemples d'appels API (Postman ou cURL)