IMT Nord Europe
École Mines-Télécom
IMT-Université de Lille

IMT Nord-Europe / FISA-TI-CI2-IN

Cloud & DevOps

Eric SIGOILLOT – 2025

Mini-projet

Le Cloud

Prise en main d'Amazon Web Services

P1.

Au fond d’un garage

En vol !

À nous la gloire ! Enfin peut-être. Faut voir.

Depuis quelques jours, ça s'agite au fond de ce garage. Charlie et Steven* discutent beaucoup. Toutes ces startups qui se lancent, et qui récoltent des millions en quelques mois… Ça donne envie !*

Ils n'ont qu'à trouver l'idée du siècle, LE truc qui attirera des millions d'utilisateurs et viendra leur remplir les poches.

Sauf que voilà, les idées, ce n'est pas leur principal atout à nos amis.

On finit vite par regarder le bout de ses pieds, et on scrolle sur son téléphone, pour faire passer le temps.

- *Hey ! On n'a qu'à refaire ~~Twitter, nan X,~~ nan Bluesky, mais en mieux.*
- *Bluesky ? Beuh, y'a rien à inventer de mieux*
- *Mais siiii, tu vas voir… Bon faut déjà se faire la main mais je le sens super bien.*
- *Mouais. Au pire ça fera passer le temps. Let's go.*
- *Au fait, euh, t'as une carte bleue ? J'ai une idée pour démarrer rapidement.*
- *…*

** Les prénoms ont été changés pour respecter leur anonymat*

En route ?

Vous voilà lancés sur l'autoroute du succès, avec comme objectif la gloire, l'argent facile et …

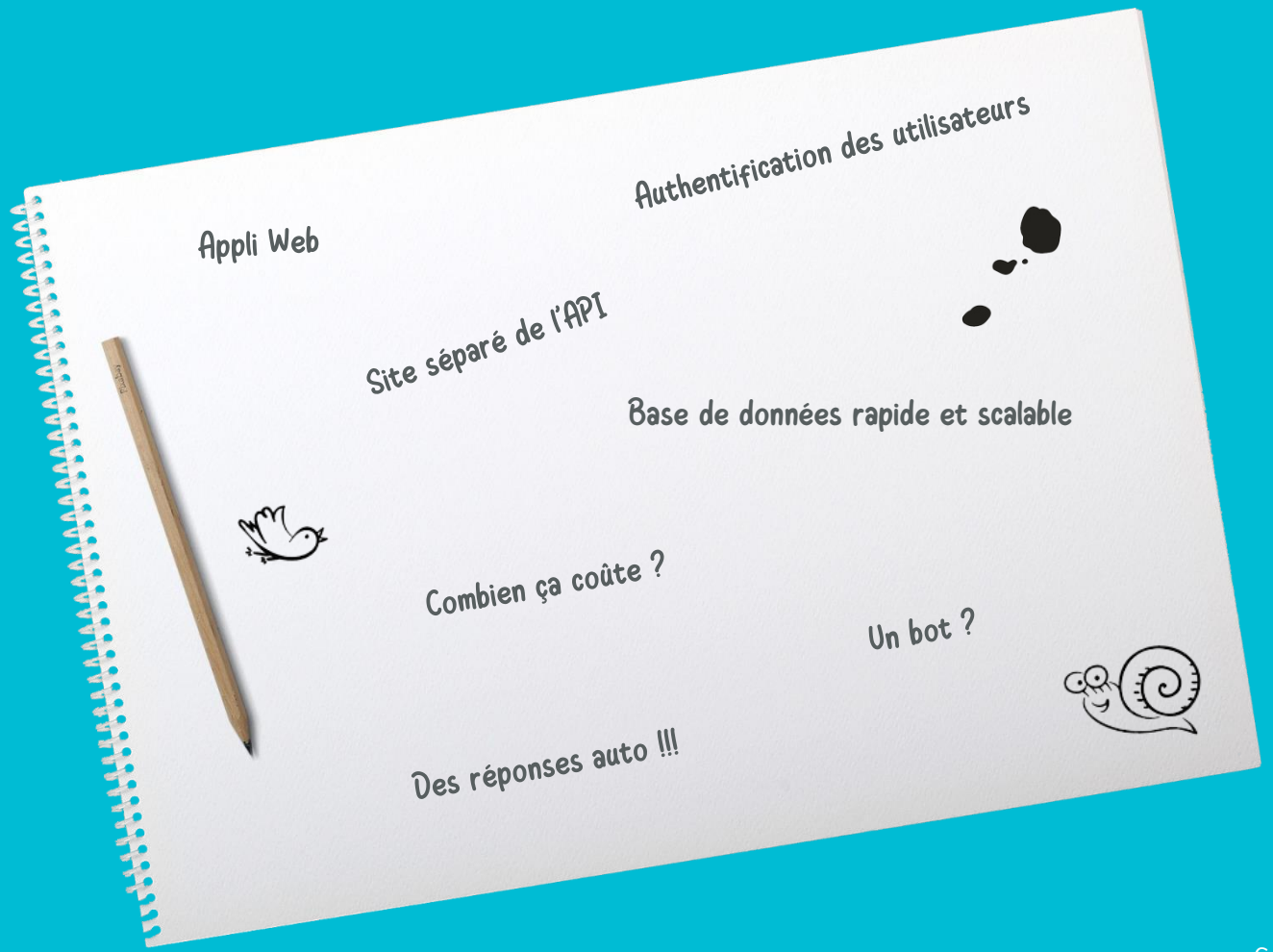
La carte bleue sortie, un compte a été créé chez le premier provider cloud qui passait par là : Charlie avait commandé un bouquin récemment, c'est donc tombé chez AWS.

Reste à se lancer pour de bon, et à… *refaire Bluesky* ?
Mouais, ça semble un poil ambitieux pour démarrer, on va peut-être viser un peu plus raisonnable.

Disons qu'on va faire une application, Web déjà, qui permet de poster des messages et d'y répondre. Ça sera déjà pas mal.

Et puis, qui sait ? On pourra aussi ajouter quelques trucs pour se faire la main.

Les idées !


Appli Web
Authentification des utilisateurs
Site séparé de l'API
Base de données rapide et scalable
Combien ça coûte ?
Un bot ?
Des réponses auto !!!

P1.

Votre mission

Début des hostilités

Votre mission, si vous l'acceptez

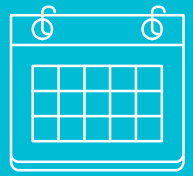
- Votre projet consiste à développer une mini-application Web accessible sur Internet
- Elle sera hébergée sur AWS, et ne devra utiliser que des services dits *serverless*.
- Afin d'être autonome rapidement, les services utilisés seront soit sur le modèle PaaS, soit sur le modèle SaaS.

- Le projet sera réalisé en binôme (sauf pour un, déso !) afin de mieux répartir les tâches, et de pouvoir expérimenter plus facilement les échanges de messages sur l'interface.
- Et puis on a souvent besoin d'un canard pour avancer.

- Le projet comprend plusieurs étapes progressives, jusqu'à des éléments facultatifs sous forme de bonus.

Le mode de fonctionnement

Vacances scolaires…



- Vous aurez environ 5 semaines à compter de la réception de ces instructions pour rendre votre projet.
  Cela signifie que vous devrez me rendre vos travaux le 18 octobre au plus tard.
- Il n'y aura pas de retard possible, pas d'excuse ou de chien qui mange vos devoirs : le 19 octobre, vos accès seront supprimés, et plus aucun changement ne sera possible.
- Vous serez notés sur plusieurs éléments :
  - La conception générale mise en place et fournie sous forme de schéma, éventuellement annoté
  - L'évaluation du pricing mensuel que vous allez faire
  - Le développement des quelques modules de code requis
- Les binômes ont été sélectionnés aléatoirement, et ne sont pas modifiables. Il faudra faire avec, comme dans la vraie vie.

Livraison

- Le rendu de votre travail s'effectuera exclusivement par email (pas de lien GitHub ou GitLab) au plus tard le lendemain de la date limite
- J'attends de votre part un mail de livraison avec au moins :
  - Un (ou plusieurs) schéma d'architecture, au format PDF ou image
  - Une évaluation de coûts argumentée, a priori sous forme de tableau, au format PDF ou Excel ou équivalent
  - Le code source de vos réalisations (Web et API) au format texte
- L'ajout d'un rapport ou autre document d'accompagnement n'est pas requis et n'apportera pas de points supplémentaires. Ce n'est pas ce que je note.

Difficultés

- Pour la très grande majorité d'entre vous, vous ne connaissez pas AWS, qu'il s'agisse de son mode de fonctionnement, de ses services ou de sa console.
- Il va donc y avoir une phase de recherche et d'analyse pour trouver de quoi vous avez besoin.
- Je vous fournis dans ce support quelques pistes pour vous permettre de ne pas trop vous perdre.
  En cas de panique, vous pouvez bien sûr me contacter pour que je vous débloque.
- Le langage de développement n'est pas imposé, mais vous devez choisir quelque chose qui convient au binôme.
  Je vous invite à utiliser Javascript ou Python, ce sera le plus simple.
  Certains de vos prédécesseurs ont tenté des langages plus exotiques et se sont plantés. Restez humbles dans vos choix :)

Sécurité

- Pour vous connecter à la console AWS, ou pour développer et déployer vos services, vous aurez à votre disposition :
  - Un numéro de compte AWS
  - Un compte personnel, avec un login et un mot de passe
  - Une région
- Vous avez interdiction de partager ces accès avec qui que ce soit d'autre
- Vous avez interdiction de déposer ces accès sur un contrôle de code source, d'autant plus public (type GitHub par exemple) où ils seraient volés
- Vous avez interdiction d'utiliser le compte AWS fourni pour toute autre chose que le mini-projet

- En cas de non-respect de ces consignes, vos accès seront supprimés immédiatement et votre note fixée à zéro.

Code source

- Pour des raisons d'équité, et le cours DevOps n'étant pas encore partagé avec vous, aucune pratique DevOps ne sera exigible pour le projet.
- Vous pouvez, si vous le souhaitez, héberger votre code source chez un provider quelconque (GitHub, GitLab, ...), sans que cela ne soit pris en compte dans la notation.

- Notez que du fait du respect des consignes de sécurité énoncées préalablement, vous ne pourrez pas automatiser le déploiement de votre solution faute d'accès à distance.

- J'insiste : vous ne devez pas placer vos accès sur un outil tel que GitHub ou GitLab.

Infrastructure as code

- Les outils d'*infrastructure as code* tels que CloudFormation ou Terraform peuvent vous aider à déployer vos ressources et les configurer.
  Toutefois, ceci représente un apprentissage en plus qui peut vous retarder.
- L'utilisation de ces outils reste à votre main, mais cela ne sera pas pris en compte dans la notation.

- Si vous souhaitez les utiliser, n'oubliez pas de ne pas intégrer les accès au cloud dans le code source et de ne jamais placer ces accès sur Git par exemple.

P1.

User stories

Qui fait quoi ?

Connexion

- En tant qu'utilisateur non authentifié, je peux accéder à l'application Web
- L'application me présente un écran de connexion sur lequel je dois renseigner mon identifiant et mon mot de passe.
- Lorsque je valide ma saisie, le système vérifie mes accès et me permet d'accéder à la partie authentifiée de l'application.
- Si mes identifiants sont rejetés, un message m'indique que mon compte est inconnu et je suis invité à recommencer.

- Une fois authentifié avec succès, si je retourne sur l'application Web sans avoir fermé mon onglet, l'application m'affiche directement la partie authentifiée de l'application sans avoir à ressaisir mes identifiants.
- Si toutefois j'ai fermé mon onglet ou mon navigateur, je devrai ressaisir mes identifiants.

Affichage des messages

- En tant qu'utilisateur authentifié, je peux consulter les messages postés par tous les utilisateurs
- Les messages s'affichent en ordre antéchronologique (le plus récent en premier) et contiennent :
  - L'identifiant du compte ayant posté le message
  - Le contenu du message

- Les messages s'affichent par paquet de 10 uniquement, et je dispose d'un moyen pour consulter les messages suivants (via une pagination ou un *scroll infini* par exemple).

Ajout de message

- En tant qu'utilisateur authentifié, je peux poster un nouveau message en cliquant sur un bouton
- Le message ne peut contenir que du texte : aucun média de type fichier, image, son ou vidéo n'est accepté.
- Une fois le message posté, il est enregistré par le système avec comme information :
  - Mon identité
  - Le contenu du message
  - La date à laquelle le message a été posté
- Une fois le message enregistré par le système, il est affiché avec les autres messages sur mon écran
- Mon nouveau message ne sera visible des autres utilisateurs que lorsqu'ils rafraichiront leur page ou se reconnecteront.

P1.

User stories bonus

Pour aller plus loin

Le bot météo

- Toutes les heures, le système ajoute automatiquement un message qui contient la météo du jour pour la ville de Lille
- Le message contiendra des informations sur la température en degrés Celsius, la couverture nuageuse et le risque de pluie.
- Ce truc pourrait servir : https://open-meteo.com/en/docs

Les réponses automatiques

- Suite à l'ajout d'un message par un utilisateur, un évènement sera émis à destination du composant de gestion des réponses automatiques.

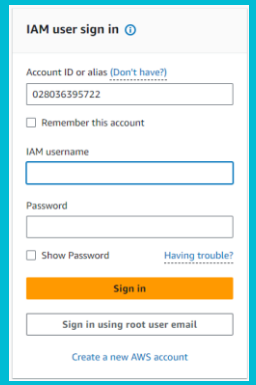
- Après analyse simple du message, des réponses automatiques pourront être ajoutées sous forme de nouveau message si des mots clés spécifiques sont détectés.

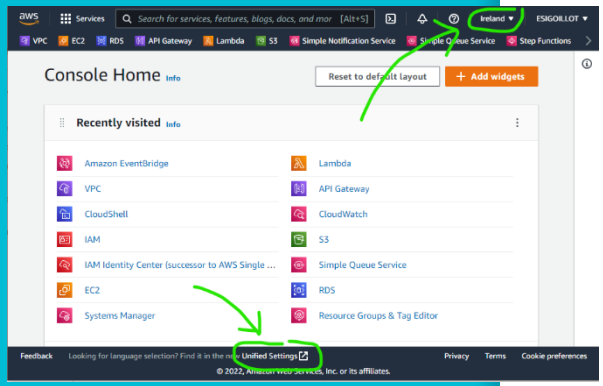
P1.

AWS

Comment démarrer ?

La console





- Pour se connecter, il faut accéder à l'adresse :

  https://028036395722.signin.aws.amazon.com/console/

- Saisissez les informations demandées (numéro de compte, login et mot de passe pour accéder à la console)
  - En cas d'accès par une autre URL, précisez que vous utilisez un compte IAM
- Une fois la console affichée, il est très fortement conseillé de la basculer en anglais : cliquez en haut à droite sur l'engrenage et changez la langue
- Vérifiez ensuite en haut à droite votre région : ce devrait être Ireland (eu-west-1) ; changez-la si nécessaire.

Vous ne retrouvez plus vos ressources ? Vérifiez votre région !

La CLI

- Vous pouvez accéder à AWS et à vos ressources directement en ligne de commande grâce à la CLI AWS
- Pour l'installer, rendez-vous ici :

https://docs.aws.amazon.com/fr_fr/cli/latest/userguide/getting-started-install.html

- L'utilisation de la CLI nécessite de disposer de clés d'accès, nommées Access Key et Secret Key. Elles vous ont été fournies avec votre compte d'accès.
- Pour les utiliser, une commande simple :

```
aws configure
```

- L'utilisation de la ligne de commande AWS n'est absolument pas requise, tout peut se faire depuis votre navigateur

Le SDK

- Pour accéder aux ressources AWS, un SDK est mis à disposition pour vos développements
- Il est disponible pour les langages principaux : **utilisez-le** pour gagner du temps, car la signature de requêtes AWS est plutôt pénible à faire à la main… Et vous n'y gagnerez rien.

- https://aws.amazon.com/fr/sdk-for-javascript/
- https://aws.amazon.com/fr/sdk-for-python/
- https://aws.amazon.com/fr/sdk-for-java/
- https://aws.amazon.com/fr/sdk-for-net/
- https://aws.amazon.com/fr/sdk-for-go/

Les tags

- Chaque ressource sur AWS peut être accompagnée de métadonnées simples, appelées des tags
- Un tag est composé d'un nom, et d'une valeur, le tout au format texte simple.
- Les tags permettent de regrouper des ressources, ou de les identifier plus précisément.
- Ils permettent aussi de définir des permissions spécifiques en fonction des tags présents sur les ressources.
- Dans votre projet, vous devrez utiliser à minima un tag sur vos ressources : le tag « `Group` », avec le nom fixé qui vous a été donné.
- Exemple : `Group = astra`
- Celui-ci permettra d'identifier vos ressources, et de leur appliquer les bonnes permissions. Vous n'avez pas mis le tag Group ? Ca ne marchera pas !

Le nommage

- Vos ressources vont toutes porter un nom unique
- Il est conseillé pour s'y retrouver de faire figurer dans le nom de la ressource son type, mais il n'y a rien d'obligatoire : à vous de voir !
- Seule obligation pour le projet : chacune de vos ressources doit contenir le nom de votre groupe **entre tirets**
- Par exemple, pour un groupe « test », un bucket S3 pourra avoir ce type de nom :

```
s3-test-site-web
```

- N'oubliez pas de mettre le nom du groupe, sinon vos actions seront interdites ! Une erreur AccessDenied ? Vérifiez le nom (et le tag Group) !

Les permissions sur les services

- Chaque ressource AWS nécessite des permissions pour faire les actions souhaitées
- Ces permissions sont attribuées par l'intermédiaire d'un rôle, sur lequel on applique une ou plusieurs stratégies (ou *permission policies*). Les permissions sont des actions sur des ressources, et sont décrites sur les stratégies.
- Un rôle est attribué, et ne peut être utilisé que par les ressources qui sont explicitement autorisées dans sa *trust policy*.

- Pour créer un rôle, vous devez donc d'abord définir sa *trust policy* (qui peut l'utiliser), et ensuite définir ses *permission policies* (ce qu'il permet de faire).
- Une fois le rôle créé, vous pourrez l'utiliser sur le service souhaité. N'oubliez pas de le taguer lui aussi !

Vos permissions

- Sur AWS, vous disposez d'un utilisateur IAM pour vous permettre de créer, modifier ou supprimer des ressources.
- Pour ce faire, un rôle vous a été attribué, et vous ne pourrez bien sûr pas en changer.
- Ce rôle vous permet d'accéder à certains types de ressources, et pas à tout ! Vous restez parqués comme un troupeau :)

- Au-delà des restrictions sur les ressources et actions, vous avez l'obligation d'appliquer des tags sur vos ressources.
- Si vous ne placez pas de tag, vos créations seront refusées !

- Vous pouvez lister toutes les ressources créées sur le cloud, mais vous ne pourrez consulter le détail que de vos propres ressources. Vous ne pourrez pas *pomper* directement sur vos voisins...

Quels services pour quel usage ?

Stocker des fichiers, dont des fichiers Web	S3
Exécuter du code en serverless	Lambda
Stocker des données en base	DynamoDB
Créer des comptes utilisateurs	Cognito User Pool
Identifier des utilisateurs et leur donner des droits	Cognito Identity Pool
Exposer une API	API Gateway
Ordonnancer des appels et envoyer des évènements	EventBridge
Définir des permissions	IAM

Restrictions

- Certains services sont restreints dans leur définition pour ne pas faire consommer inutilement des ressources :
  - Amazon S3 : pas de versioning sur les fichiers
  - Lambda : 128 Mo par fonction, timeout maximum de 5 secondes
  - DynamoDB : capacité provisionnée avec 1 RCU et 1 WCU
  - API Gateway : API REST régionale exclusivement

- Vous n'aurez pas de nom de domaine personnalisé, et les certificats seront gérés directement par AWS
- Vous ne mettrez pas en place de solution de sécurité avancée

Toute ressource créée doit être payée : pensez à ne créer que le strict nécessaire, et à supprimer ce qui ne sert pas.
La surconsommation est mauvaise pour les finances, et pour la planète.

P1.

Pistes et aide au démarrage

Dans le grand bain

Tutoriels

- Vous trouverez de nombreux tutoriels sur Internet pour démarrer avec AWS
  - Ne tentez pas de créer les ressources comme ils vous le proposent : vous n'aurez probablement pas le droit de le faire à l'identique
  - Tentez plutôt de comprendre ce qu'ils font faire, plutôt que de le faire directement

- Le mini-projet n'est pas basé sur un tutoriel existant : il a été inventé pour le besoin du cours
  - Ne cherchez pas un tutoriel qui correspond exactement, et utilisez plutôt votre cerveau : il semble fiable, sinon vous ne seriez pas là ☺

Exposer une API

- Vous utiliserez sûrement une API Gateway avec une API REST
- L'API doit être exposée sur Internet, mais on ne vise pas le monde entier non plus
- L'API ne doit pas être ouverte aux quatre vents : vous devez la sécuriser avec un mécanisme spécifique pour autoriser les accès
- Il doit être possible de brancher une API avec une ou plusieurs fonctions serverless…
- Qui dit API, dit contrat : vous devriez le définir précisément
- Une API Gateway définit des ressources, qui sont ensuite déployées sur un stage
- Vous avez changé quelque chose ? Pensez à redéployer l'API !

Exposer une API (suite)

- SI vous utilisez la console, vous ne pourrez pas appliquer de tag sur l'API Gateway lors de sa création : ça sera autorisé
- Par contre, toutes les modifications par la suite seront refusées si l'API Gateway n'est pas taguée !
- Après sa création, pensez à accéder à la partie Settings, puis Configure Tags

- L'API Gateway aura son propre nom de domaine, différent de celui du site Web : attention à la problématique CORS !

Exposer un site Web

- Les buckets S3 peuvent être transformés en sites Web en quelques clics : mais ce sont des sites HTTP uniquement
- Si vous voulez un site HTTPS, il faudra passer par un autre service pour l'exposer
- CloudFront est un service complexe, et qui vous fera attendre longtemps entre chaque changement de configuration : à vous de voir si c'est le plus pertinent
- On peut aussi utiliser une API Gateway et une fonction Lambda pour aller chercher les fichiers

- Dans tous les cas, évitez de bloquer tous les accès publics au bucket, sinon ça sera compliqué d'y accéder !
- Pensez également qu'un bucket possède sa propre policy pour autoriser ou bloquer les accès.

Exposer un site Web (suite)

- Si vous utilisez la console, vous ne pourrez pas appliquer de tag sur une distribution CloudFront lors de sa création : ça sera autorisé
- Par contre, toutes les modifications par la suite seront refusées si la distribution n'est pas taguée !
- Après sa création, pensez à la modifier et à ajouter le tag Group requis dans l'onglet Tags

- Vous serez amenés à choisir une zone pour CloudFront : la zone Etats-Unis/Europe devrait être suffisante

Stocker des données

- DynamoDB fonctionne à partir de tables : chaque table contient des données spécifiques
- Si vous regardez bien, il n'est pas impossible qu'une table DynamoDB prête à accueillir vos « *skeets* » soit déjà disponible. Quelle coïncidence !

- Vous ne serez pas autorisés à faire des scans sur les tables : ils sont coûteux, lents, et synonymes de mauvais choix d'index

Exécuter une fonction serverless

- Les fonctions Lambda ont besoin d'un rôle pour s'exécuter
- Il faudra créer un rôle personnalisé, et lui attribuer a minima la policy nommée « AWSLambdaBasicExecutionRole » pour lui permettre de s'exécuter seule
- Si elle a besoin d'accéder à d'autre chose… il faudra ajouter d'autres policies, ou les créer vous-même

- Une fonction appelée par une API Gateway doit avoir une sortie spécifique pour être utilisable : jetez un œil à la documentation
- L'intégration API Gateway / Lambda se fera vraisemblablement en mode proxy

Gérer des utilisateurs

- Cognito est le service idéal pour gérer des utilisateurs
- Il est scindé en deux parties : User Pool, pour les utilisateurs, et Identity Pool, pour les permissions
- Cognito permet d'attribuer des permissions sur des ressources AWS à un utilisateur authentifié : pratique, il s'interface a priori avec le service API Gateway…
- Inutile de redévelopper des écrans de connexion : Cognito fournit des choses clés en main. Un bon plan !

Les évènements

- EventBridge permet de réagir à des évènements : soit ceux émis par les services AWS, soit ceux qu'on lui envoie, tout simplement
- Il possède aussi des capacités d'ordonnancement : donc de planification de tâches

Les permissions

- N’oubliez pas que chaque service doit être autorisé à faire les actions souhaitées, et uniquement les actions souhaitées
- Si vous n’affectez pas de rôles à vos services, ils ne pourront rien faire par eux-mêmes
- Certains services ont également besoin de définir qui est autorisé à les appeler : c’est notamment le cas des lambdas avec les *lambdas permissions*
- Si vous branchez une API Gateway sur une lambda, il faudra donc autoriser l’API Gateway à appeler la lambda…
- Vous devez taguer toutes vos ressources : vous ne pourrez donc pas utiliser les fonctions de création automatique de rôle sur la console AWS, désolé !

Last but not least…

Ce projet a été confié à l'identique à vos prédécesseurs les années passées.

Je possède toujours leur code source, et je comparerai vos réalisations avec les leurs.
Évitez la copie ou l'inspiration trop franche, elle est mauvaise pour la santé de vos notes.

ChatGPT est un outil fantastique ; il devrait pouvoir réaliser ce projet en quelques prompts bien réglés et en moins d'une heure, café compris.
Problème : vous êtes a priori bien moins doués que lui, et ça va donc se voir fortement…
Si vous souhaitez faire appel à cet ami, il n'y a pas de contre-indication, mais faites-le intelligemment, car c'est votre code que je note (entre autres choses…), pas celui de l'IA !

Ne passez pas trop de temps sur l'habillage ou le look de l'ensemble. Ma grille de notation s'appuie sur les fonctionnalités qui marchent. Ce n'est pas un cours de CSS ou de marketing.

En cas de binôme avec un poil dans la main façon cocotier, je tolère les dénonciations calomnieuses et lettres de corbeau. Mais c'est moche, je vous conseille plutôt la motivation mutuelle.

Bon courage à vous !

Et n’oubliez pas de m’envoyer votre schéma d’architecture, et votre évaluation de pricing :)