**Choix de la base de données**

* **Type de base de données**

- Relationnel

- NoSQL

* **Justification des choix**
* **Pour les assets et les factures :**

[**MySQL**](https://www.mysql.com/)**:** On utilise une base de données relations, pour les assets et les factures. Simple d’utilisation, facile à mettre en place, on n’a pas besoin de faire un énorme scaling au début, étant donné que les assets ne vont pas trop bouger et les factures n’ont pas besoin d’être accessibles dans l’immédiat. On s’est décidé sur MySQL, puisqu’elle est populaire et on peut mettre en place système d’administration sur les bases de données ([phpMyAdmin](https://www.phpmyadmin.net/)) ainsi qu'un orchestrateur pour les réplications des bases de données ([orchestrator](https://github.com/openark/orchestrator)). Cette dernière va être sur un [serveur dédié](https://fr.wikipedia.org/wiki/Serveur_d%C3%A9di%C3%A9) (pas besoin de faire d’adapter le [scaling](https://blog.developpez.com/sqlpro/p10383/langage-sql-norme/sgbdr_et_repartion_de_charge_scalabilite#:~:text=La%20%C2%AB%20scalabilit%C3%A9%20%C2%BB%20(en%20fran%C3%A7ais,permettre%20une%20r%C3%A9partition%20de%20charge.) du serveur). Les deux bases de données seront hébergées sur un serveur pour des soucis de budget, et auront une réplication sur un VPS qui n’a pas de grosse performance mais qui est là uniquement pour des soucis de sécurité des données. On peut passer par OVH pour le serveur dédié loué au mois pour une durée de 1an, ce qui revient dans [les 1100€ environ](https://www.ovhcloud.com/fr/bare-metal/advance/adv-apac/). En plus de ce serveur, on peut prendre un VPS (serveur privé virtuel) sur Hostinger pour avoir une réplication des données sur autre serveur. Ce dernier n’est pas dans la même région et n’est pas de la même entreprise pour éviter qu'un incident qui touche une entreprise n'impacte pas les autres serveurs. Cela revient à [environ 200€](https://cart.hostinger.com/pay/27413e95-138d-4bc6-b6d6-4f43563514f4) pour le VPS, ce qui nous fait un total de 1300€ pour les bases de données SQL.

* **Pour les sauvegardes des utilisateurs :**

[**Apache Cassandra**](https://cassandra.apache.org/_/index.html)**:** On va préférer une base de données NoSQL pour les sauvegarde des utilisateurs pour l'accessibilité et la disponibilité, qui est crucial pour notre jeu. Contrairement à une base de données SQL, cette dernière doit pouvoir être performante, disponible tout le temps et à tout temps. Cassandra a l'avantage [de gérer la moindre erreur](https://www.datastax.com/blog/cassandra-error-handling-done-right), pour éviter la perte de données de sauvegarde des utilisateurs, ce qui peut être frustrant pour le joueur. Étant donné que l’on peut faire évoluer notre demande pour ce qui est des sauvegardes de l’utilisateur, on décidera plutôt de passer par AWS pour pouvoir adapter nos besoins sur les serveurs. On peut décider qu’il nous faudra un peu près 300Go de stockage par an pour un peu près [1500€ environ](https://aws.amazon.com/keyspaces/pricing/), incluant un backup en cas de besoin des données.

* **Prix des services d’hébergement cloud “classique” :**

Dans l’avenir, si on souhaite partir sur un serveur plus adaptable pour nos bases de données relationnels, on pourrait partir sur du AWS et Azure.

|  |
| --- |
| **Prix de AWS pour Aurora MySQL** |

|  |
| --- |
| **Prix pour Azure SQL** |

Cependant, cela prend en compte des options comme la surcharge du serveur, ainsi que d'autres services que l’on ne risque pas d’utiliser dans l'immédiat. De ce fait, on va plutôt préférer un VPS ou bien un serveur dédié pour héberger nos bases de données SQL.