

OC PIZZA



SOLUTION DE GESTION DE PRISE DE COMMANDE VIA UNE APPLICATION ANDROID

Dossier d'exploitation

Version 1.1

Dossier réf. : **DE-ITCD20230323OCP**

Auteur

HAMMER JULIEN

Lead développeur

TABLE DES MATIÈRES

1 - Versions.....	3
2 - Introduction	4
2.1 - Objet du document	4
2.2 - Références.....	4
3 - Pré-requis	5
3.1 - Système.....	5
3.1.1 - Serveur de base de données.....	5
3.1.2 - Caractéristiques techniques.....	6
3.2 - Bases de données.....	7
3.2.1 - Modèle relationnel :.....	7
3.2.2 - Scripts SQL creation de la base de données :.....	8
3.2.3 - Scripts SQL insertion dans la base de données :.....	10
3.3 - Web-services.....	13
4 - Procédure de déploiement	14
4.1 - Déploiement de la base de données.....	14
4.1.1 - Base de données et outil d'administration.....	14
4.1.2 - Procédure de connexion avec MySQL Workbench.....	14
4.1.3 - Vérifications.....	17
5 - Procédure de démarrage / arrêt	18
5.1 - Avantages.....	18
5.2 - Limites.....	18
5.3 - Arrêt temporaire.....	19
6 - Procédure de mise à jour	20
7 - Supervision/Monitoring	21
7.1.1 - Rapports de performance prêts à l'emploi.....	24
8 - Procédure de sauvegarde et restauration	25
9 - Ressources	26
10 - Glossaire	27

1 - VERSIONS

Auteur	Date	Description	Version
HAMMER JULIEN	23/03/2023	Création du document	1.1
HAMMER JULIEN	24/03/2023	Finalisation du document	1.1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier d'exploitation de l'application android dans le but de remplacer l'ancien système informatique du groupe OC PIZZA devenu obsolète.

L'objectif de ce document est donc de présenter les fonctionnalités du nouveau système informatique.

Les éléments du présent dossier découlent :

- de la demande initiale du client Franck et Lola d'OC Pizza,
- des différents échanges avec ce client.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1 - DCT – réf. DCT-ITCD20230323OCP : Dossier de conception technique de l'application

2 - DCF – réf. DCF-ITCD20230323OCP : Dossier de conception fonctionnelle de l'application

3 - PRÉ-REQUIS

3.1 - Système

3.1.1 - Serveur de base de données

Le serveur de base de données sera hébergé chez Amazon RDS pour MySQL. Une fois le serveur créé il est possible de connecter MySQL d'AWS RDS avec l'outil MySQL Workbench.

BASE DE DONNÉES

Offre gratuite 12 MOIS GRATUITS

Amazon RDS

750 heures

par mois d'utilisation de base de données
(moteurs de base de données applicables)

Service de base de données
relationnelle géré pour MySQL,
PostgreSQL, MariaDB ou SQL Server.

750 heures d'utilisation des instances Amazon
RDS Single-AZ db.t2.micro, db.t3.micro et
db.t4g.micro avec des bases de données
MySQL, MariaDB, PostgreSQL chaque mois
(moteurs de base de données applicables)

20 Go de stockage de base de données à
usage général (SSD)

20 Go de stockage pour les sauvegardes et
instantanés de base de données

3.1.2 - Caractéristiques techniques

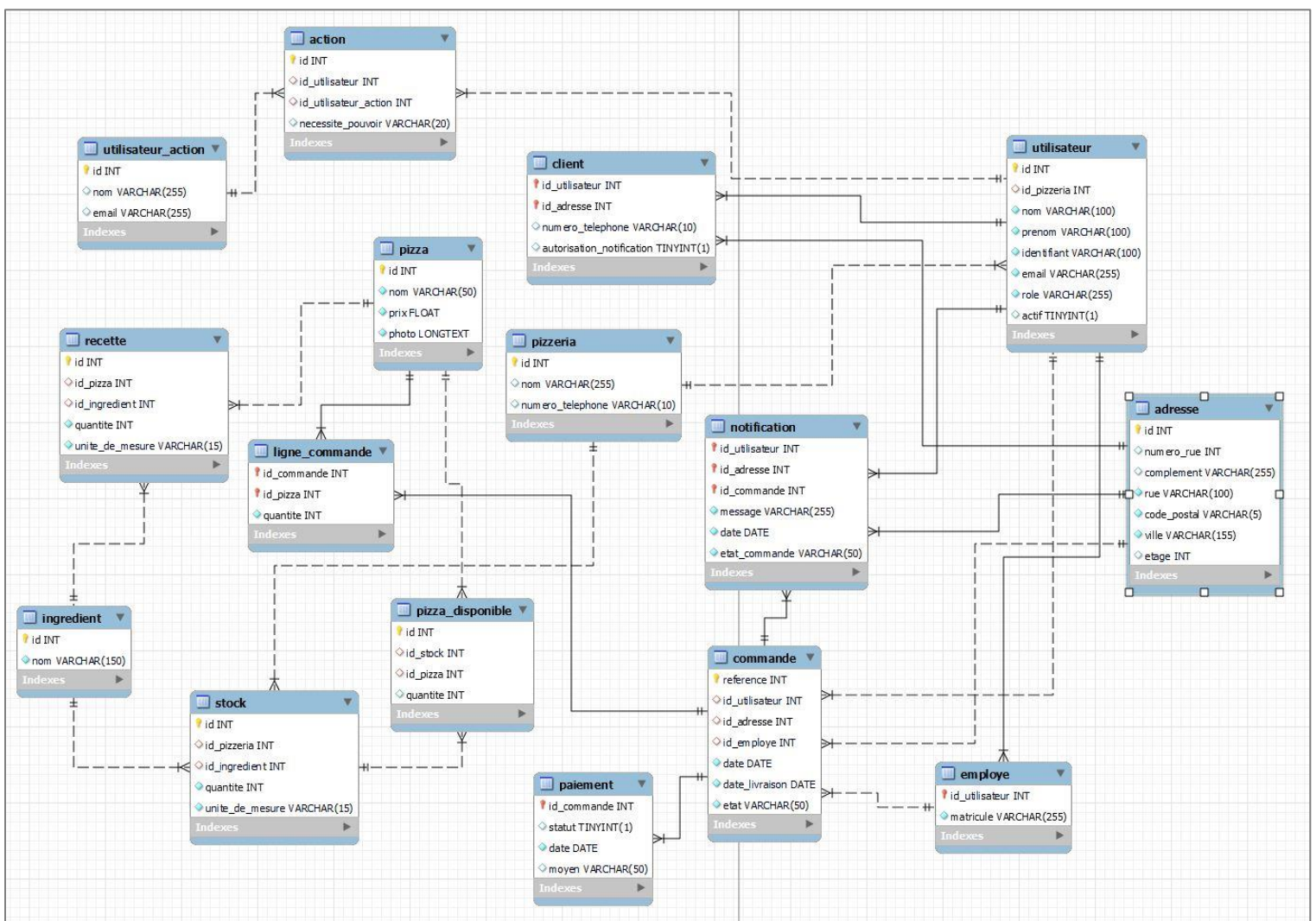
- Avec Amazon RDS, vous pouvez déployer des serveurs MySQL évolutifs en quelques minutes avec une capacité matérielle économique et redimensionnable.
- Amazon RDS for MySQL vous permet de concentrer vos efforts sur le développement des applications en prenant en charge les tâches chronophages d'administration de bases de données, notamment les sauvegardes, les mises à niveau, les corrections de logiciels, l'amélioration des performances, la surveillance, ou encore la mise à l'échelle et la réplication.
- Amazon RDS prend en charge MySQL Community Edition versions 5.7 et 8.0, ce qui signifie que le code, les applications et les outils peuvent être utilisés avec Amazon RDS.

3.2 - Bases de données

Les bases de données et schémas suivants doivent être accessibles et à jour :

- **Modèle relationnel de la base de données et script creation et insertion de données :** version 1.

3.2.1 - Modèle relationnel :



3.2.2 - Scripts SQL creation de la base de données :

```

1 • CREATE database OCPizza;
2 • use OCPizza;
3
4 • CREATE TABLE adresse (
5     id INTEGER AUTO_INCREMENT PRIMARY KEY,
6     numero_rue INTEGER,
7     complement VARCHAR(255),
8     rue VARCHAR(100) NOT NULL,
9     code_postal VARCHAR(5) NOT NULL,
10    ville VARCHAR(155) NOT NULL,
11    etage INTEGER
12 );
13
14 • CREATE TABLE utilisateur_action(
15     id INTEGER AUTO_INCREMENT PRIMARY KEY,
16     nom VARCHAR(255),
17     email VARCHAR(255)
18 );
19
20 • CREATE TABLE pizzeria(
21     id INTEGER AUTO_INCREMENT PRIMARY KEY,
22     nom VARCHAR(255),
23     numero_telephone VARCHAR(10)
24 );
25
26 • CREATE TABLE utilisateur (
27     id INTEGER AUTO_INCREMENT PRIMARY KEY,
28     id_pizzeria INTEGER,
29     nom VARCHAR(100) NOT NULL,
30     prenom VARCHAR(100) NOT NULL,
31     identifiant VARCHAR(100) NOT NULL,
32     email VARCHAR(255) NOT NULL,
33     role VARCHAR(255) NOT NULL,
34     actif BOOLEAN DEFAULT TRUE,
35     FOREIGN KEY (id_pizzeria) REFERENCES pizzeria(id)
36 );

```

```

37
38 • CREATE TABLE employe(
39     id_utilisateur INTEGER,
40     matricule VARCHAR(255) NOT NULL,
41     PRIMARY KEY (id_utilisateur),
42     FOREIGN KEY (id_utilisateur) REFERENCES utilisateur(id)
43 );
44
45 • CREATE TABLE commande (
46     reference INTEGER AUTO_INCREMENT PRIMARY KEY,
47     id_utilisateur INTEGER,
48     id_adresse INTEGER,
49     id_employe INTEGER,
50     date DATE NOT NULL,
51     date_livraison DATE NOT NULL,
52     etat VARCHAR(50) NOT NULL,
53     FOREIGN KEY (id_utilisateur) REFERENCES utilisateur(id),
54     FOREIGN KEY (id_adresse) REFERENCES adresse(id),
55     FOREIGN KEY (id_employe) REFERENCES employe(id_utilisateur)
56 );
57
58 • CREATE TABLE notification (
59     id_utilisateur INTEGER,
60     id_adresse INTEGER,
61     id_commande INTEGER,
62     message VARCHAR(255) NOT NULL,
63     date DATE NOT NULL,
64     etat_commande VARCHAR(50) NOT NULL,
65     PRIMARY KEY (id_utilisateur, id_adresse, id_commande),
66     FOREIGN KEY (id_utilisateur) REFERENCES utilisateur(id),
67     FOREIGN KEY (id_adresse) REFERENCES adresse(id),
68     FOREIGN KEY (id_commande) REFERENCES commande(reference)
69 );
70

```



```

70
71 ● CREATE TABLE paiement(
72     id_commande INTEGER,
73     statut BOOLEAN DEFAULT FALSE,
74     date DATE NOT NULL,
75     moyen VARCHAR(50),
76     PRIMARY KEY(id_commande),
77     FOREIGN KEY (id_commande) REFERENCES commande(reference)
78 );
79
80 ● CREATE TABLE client(
81     id_utilisateur INTEGER,
82     id_adresse INTEGER,
83     numero_telephone VARCHAR(10),
84     autorisation_notification BOOLEAN DEFAULT FALSE,
85     PRIMARY KEY (id_utilisateur, id_adresse),
86     FOREIGN KEY (id_utilisateur) REFERENCES utilisateur(id),
87     FOREIGN KEY (id_adresse) REFERENCES adresse(id)
88 );
89
90 ● CREATE TABLE pizza(
91     id INTEGER AUTO_INCREMENT PRIMARY KEY,
92     nom VARCHAR(50) NOT NULL,
93     prix FLOAT NOT NULL,
94     photo LONGTEXT NOT NULL
95 );
96
97 ● CREATE TABLE ingredient(
98     id INTEGER AUTO_INCREMENT PRIMARY KEY,
99     nom VARCHAR(150) NOT NULL
100 );
101

```

```

102 ● CREATE TABLE stock(
103     id INTEGER AUTO_INCREMENT PRIMARY KEY,
104     id_pizzeria INTEGER,
105     id_ingredient INTEGER,
106     quantite INTEGER NOT NULL,
107     unite_de_mesure VARCHAR(15) NOT NULL,
108     FOREIGN KEY (id_pizzeria) REFERENCES pizzeria(id),
109     FOREIGN KEY (id_ingredient) REFERENCES ingredient(id)
110 );
111
112 ● CREATE TABLE recette(
113     id INTEGER AUTO_INCREMENT PRIMARY KEY,
114     id_pizza INTEGER,
115     id_ingredient INTEGER,
116     quantite INTEGER NOT NULL,
117     unite_de_mesure VARCHAR(15) NOT NULL,
118     FOREIGN KEY (id_pizza) REFERENCES pizza(id),
119     FOREIGN KEY (id_ingredient) REFERENCES ingredient(id)
120 );
121
122 ● CREATE TABLE ligne_commande(
123     id_commande INTEGER,
124     id_pizza INTEGER,
125     quantite INTEGER NOT NULL,
126     PRIMARY KEY (id_commande, id_pizza),
127     FOREIGN KEY (id_commande) REFERENCES commande(reference),
128     FOREIGN KEY (id_pizza) REFERENCES pizza(id)
129 );
130
131 ● CREATE TABLE pizza_disponible(
132     id INTEGER AUTO_INCREMENT PRIMARY KEY,
133     id_stock INTEGER,
134     id_pizza INTEGER,
135     quantite INTEGER DEFAULT 0,
136     FOREIGN KEY (id_stock) REFERENCES stock(id),
137     FOREIGN KEY (id_pizza) REFERENCES pizza(id)
138 );
139
140 ● CREATE TABLE action(
141     id INTEGER AUTO_INCREMENT PRIMARY KEY,
142     id_utilisateur INTEGER,
143     id_utilisateur_action INTEGER,
144     necessite_pouvoir VARCHAR(20),
145     FOREIGN KEY (id_utilisateur) REFERENCES utilisateur(id),
146     FOREIGN KEY (id_utilisateur_action) REFERENCES utilisateur_action(id)
147 );
148

```

3.2.3 - Scripts SQL insertion dans la base de données :

```
34
35 • INSERT INTO commande (id_utilisateur, id_adresse, id_employe, date, date_livraison, etat)
36 VALUES
37 (1,1,4,'2023-03-09', '2023-03-09', 'Commande non traitée'),
38 (1,1,4,'2023-03-09', '2023-03-09', 'Commande en cours de preparation'),
39 (1,1,5,'2023-03-09', '2023-03-09', 'Commande en cours de livraison');
40
41 • INSERT INTO notification (id_utilisateur, id_adresse, id_commande, message, date, etat_commande)
42 VALUES
43 (1,1,1,'La situation de votre commande est : ', '2023-03-09', 'Commande non traitée'),
44 (1,1,2,'La situation de votre commande est : ', '2023-03-09', 'Commande en cours de preparation'),
45 (1,1,3,'La situation de votre commande est : ', '2023-03-09', 'Commande en cours de livraison');
46
47 • INSERT INTO paiement (id_commande, statut, date, moyen)
48 VALUES
49 (1, true, '2023-03-09', 'Carte bancaire');
50
51 • INSERT INTO client(id_utilisateur, id_adresse, numero_telephone, autorisation_notification)
52 VALUES
53 (1, 1, '0303030303', true),
54 (2, 2, '0404040404', false),
55 (3, 3, '0505050505', true);
56
57 • INSERT INTO pizza (nom, prix, photo)
58 VALUES
59 ('Margherita', 6.50, 'https://www.istockphoto.com/fr/photos/pizza'),
60 ('Végétarienne', 9.50, 'https://www.istockphoto.com/fr/photos/pizza');
61
62 • INSERT INTO ingredient (nom)
63 VALUES
64 ('pâte à pizza'),
65 ('sauce tomate'),
66 ('origan');
67
```

```
67
68 • INSERT INTO stock (id_pizzeria, id_ingredient, quantite, unite_de_mesure)
69 VALUES
70 (1, 1, 100, 'g'),
71 (1, 2, 500, 'ml'),
72 (1, 3, 50, 'g');
73
74 • INSERT INTO recette (id_pizza, id_ingredient, quantite, unite_de_mesure)
75 VALUES
76 (1, 1, 2, 'g'),
77 (2, 2, 1, 'ml'),
78 (1, 3, 5, 'g');
79
80 • INSERT INTO ligne_commande (id_commande, id_pizza, quantite)
81 VALUES
82 (1, 1, 2),
83 (1, 2, 3);
84
85 • INSERT INTO pizza_disponible (id_stock, id_pizza, quantite)
86 VALUES
87 (1, 1, 5),
88 (2, 2, 5);
89
90 • INSERT INTO action ( id_utilisateur, id_utilisateur_action, necessite_pouvoir)
91 VALUES
92 (4, 3, 'EMPLOYE'),
93 (5, 4, 'EMPLOYE'),
94 (6, 5, 'ADMIN');
95
96
```



```
1 • INSERT INTO adresse (numero_rue, complement, rue, code_postal, ville, etage)
2 VALUES
3 (15, "entrée à l'arrière", "rue des jardins", "67450", "Mundolsheim", 1),
4 (2, "", "avenue des Lilas", "75019", "Paris", 3),
5 (1, "", "rue du Général de Gaulle", "59000", "Lille", 2);
6
7 • INSERT INTO utilisateur_action (nom, email)
8 VALUES
9 ('Deniro Robert', 'robertdeniro@gmail.com'),
10 ('Dujardin Jean', 'jeandujardin@gmail.com'),
11 ('Dupont Alice', 'alicedupont@gmail.com'),
12 ('Martin Julien', 'julienmartin@gmail.com'),
13 ('Meyer Lola', 'lolameyer@gmail.com'),
14 ('Meyer Franck', 'franckmeyer@gmail.com');
15
16 • INSERT INTO pizzeria (nom, numero_telephone)
17 VALUES
18 ('Chez Franck', '0323456789');
19
20 • INSERT INTO utilisateur (id_pizzeria, nom, prenom, identifiant, email, role, actif)
21 VALUES
22 (1, 'Deniro', 'Robert', 'Roro', 'robertdeniro@gmail.com', 'CLIENT', true),
23 (1, 'Dujardin', 'Jean', 'Jd', 'jeandujardin@gmail.com', 'CLIENT', true),
24 (1, 'Pitt', 'Brad', 'Bratt', 'bradpitt@gmail.com', 'CLIENT', true),
25 (1, 'Dupont', 'Alice', 'Aly', 'alicedupont@gmail.com', 'PIZZAIOLO', true),
26 (1, 'Martin', 'Julien', 'Juju', 'julienmartin@gmail.com', 'LIVREUR', true),
27 (1, 'Meyer', 'Lola', 'Lolo', 'lolameyer@gmail.com', 'ADMIN', true),
28 (1, 'Meyer', 'Franck', 'Fran', 'franckmeyer@gmail.com', 'ADMIN', true);
29
30 • INSERT INTO employe (matricule, id_utilisateur)
31 VALUES
32 ('PCF1', 4),
33 ('LCF1', 5);
```

Tous les scripts sont en pièce jointe en format zip.

3.3 - Web-services

Les web services suivants doivent être accessibles et à jour :

Paielement :



Stripe : <https://stripe.com/>

4 - PROCÉDURE DE DÉPLOIEMENT

4.1 - Déploiement de la base de données

4.1.1 - Base de données et outil d'administration

1. La base de données est hébergée chez Amazon RDS et de type MySQL en version 5.7 et 8.0.
2. Il faudra ensuite établir la connexion avec la base de données et MySQL Workbench pour une utilisation simple et intuitive.
3. Pensez à bien conserver les informations requises (nom d'hôte, port et nom d'utilisateur) pour la connexion entre Amazon RDS et MySQL Workbench.

4.1.2 - Procédure de connexion avec MySQL Workbench

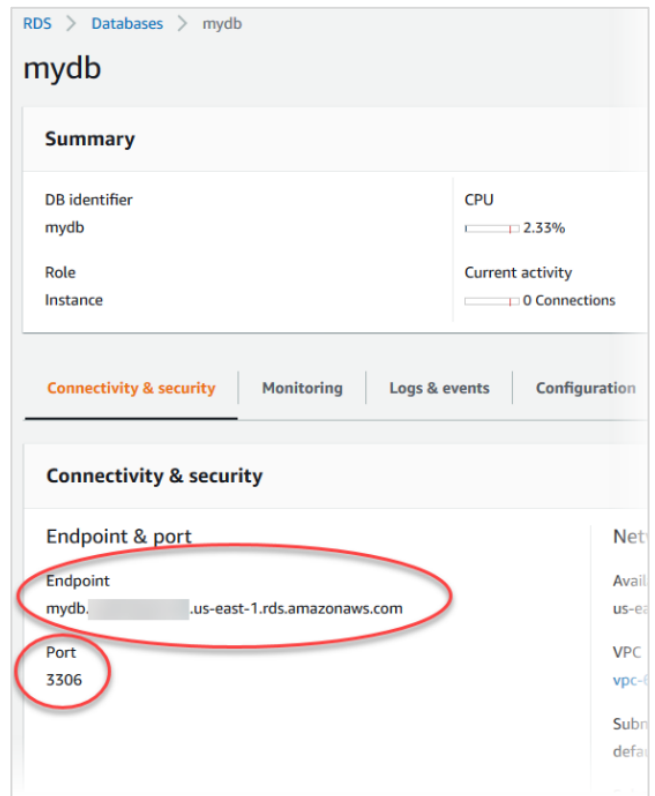
1. **Téléchargez et installez MySQL Workbench sur :**
<https://dev.mysql.com/downloads/workbench/>
2. **Ouvrez MySQL Workbench**, ensuite, **choisissez le signe ⊕** à côté de MySQL Connections, afin de configurer une nouvelle connexion.
3. Dans la boîte de dialogue **Configurez une nouvelle connexion**, ensuite, **entrez un nom** approprié pour votre connexion.
4. Dans la section **Paramètres**, saisissez les informations suivantes :
 - **Nom d'hôte** : saisissez le point de terminaison RDS.
 - **Port** : saisissez le numéro de port.
 - **Nom d'utilisateur** : saisissez l'utilisateur principal

Remarque : vous pouvez obtenir ces informations à partir de la console Amazon RDS. Dans la section **Bases** de données, choisissez **Instances**, ensuite, sélectionnez l'instance à laquelle vous vous connectez. Dans l'onglet **Connectivité et sécurité**, cliquez sur Configuration. L'utilisateur principal est répertorié ici.

Si vous recherchez le nom d'utilisateur principal, choisissez l'onglet **Configuration** et affichez la valeur Master **username (Identifiant principal)**.

1. Choisissez **Tester la connexion**.
2. Lorsqu'une fenêtre contextuelle apparaît, entrez le mot de passe configuré lors de la création de l'instance des bases de données, ensuite, choisissez **OK**.
3. Après le test de votre connexion, dans la boîte de dialogue **Configurer une nouvelle connexion**, ensuite, choisissez **OK** afin d'enregistrer la connexion.

Si vous recevez une erreur lorsque vous testez la connexion, vérifiez que vous avez configuré le réseau correctement.



RDS > Databases > mydb

mydb

Summary

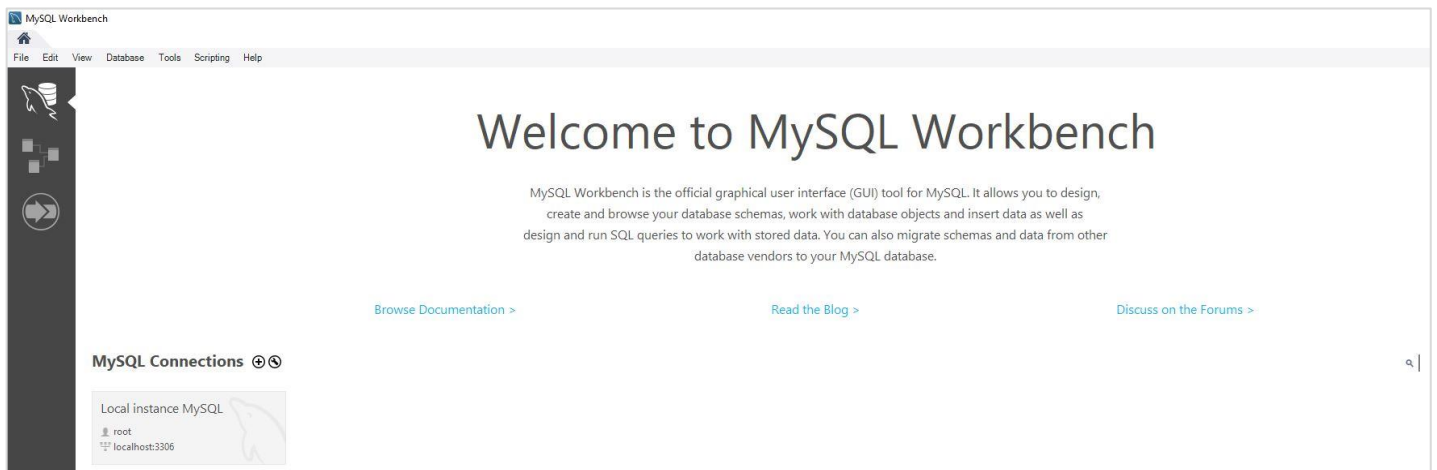
DB identifier mydb	CPU 2.33%
Role Instance	Current activity 0 Connections

Connectivity & security | Monitoring | Logs & events | Configuration

Endpoint & port

Endpoint
mydb. .us-east-1.rds.amazonaws.com

Port
3306



MySQL Workbench

File Edit View Database Tools Scripting Help

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

[Browse Documentation >](#) [Read the Blog >](#) [Discuss on the Forums >](#)

MySQL Connections ⓘ ⓘ

Local instance MySQL

root
localhost:3306



MySQL Workbench

Local instance MySQL x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

ocpizza

- Tables
 - action
 - adresse
 - client
 - commande
 - employe
 - ingredient
 - ligne_commande
 - notification
 - paiement
 - pizza
 - pizza_disponible
 - pizzeria
 - recette
 - stock
 - utilisateur
 - utilisateur_action
- Views
- Stored Procedures
- Functions

sys

Administration Schemas

Information

No object selected

Object Info Session

hammer_julien_3_scripts.sql_0...

Limit to 1000 rows

```
1 -- CREATE the database
2
3 CREATE database OCPizza;
4
5 -- USE the database to modify
6
7 use OCPizza;
8
9 -- CREATE all tables
10
11 CREATE TABLE adresse (
12     id INTEGER AUTO_INCREMENT PRIMARY KEY,
13     numero_rue INTEGER,
14     complement VARCHAR(255),
15     rue VARCHAR(100) NOT NULL,
16     code_postal VARCHAR(5) NOT NULL,
17     ville VARCHAR(155) NOT NULL,
18     etage INTEGER
19 );
20
21 CREATE TABLE utilisateur_action(
22     id INTEGER AUTO_INCREMENT PRIMARY KEY,
23     nom VARCHAR(255),
24     email VARCHAR(255)
25 );
26
27 CREATE TABLE pizzeria(
28     id INTEGER AUTO_INCREMENT PRIMARY KEY,
29     nom VARCHAR(255),
30     numero_telephone VARCHAR(10)
31 );
32
33 CREATE TABLE utilisateur (
34     id INTEGER AUTO_INCREMENT PRIMARY KEY,
35     id_pizzeria INTEGER,
36     nom VARCHAR(100) NOT NULL,
```

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
---	------	--------	---------	------------------

4.1.3 - Vérifications

1. **Vérifier que la base de données est disponible** : Vous devez vous assurer que la base de données est disponible et accessible à partir de l'application Android. Vous pouvez utiliser des outils tels que MySQL Workbench ou des commandes SQL pour vérifier la disponibilité de la base de données.
2. **Vérifier les autorisations de la base de données** : Vous devez vous assurer que les autorisations de la base de données sont correctement configurées et que les utilisateurs autorisés peuvent accéder aux données. Vous pouvez utiliser les paramètres de sécurité d'Amazon RDS pour configurer les autorisations de la base de données.
3. **Vérifier la configuration de la base de données** : Vous devez vous assurer que la base de données est correctement configurée pour l'application Android. Vous pouvez vérifier les paramètres de la base de données tels que les tables, les colonnes, les index et les clés étrangères pour vous assurer qu'ils sont correctement définis.
4. **Vérifier la performance de la base de données** : Vous devez surveiller les performances de la base de données et vérifier que les temps de réponse sont raisonnables et que la base de données ne présente pas de goulot d'étranglement. Vous pouvez utiliser des outils tels que MySQL Workbench ou des outils de surveillance de la performance d'Amazon RDS pour surveiller les performances de la base de données.
5. **Vérifier la sécurité de la base de données** : Vous devez vous assurer que la base de données est sécurisée contre les attaques. Vous pouvez vérifier les paramètres de sécurité d'Amazon RDS pour vous assurer que la base de données est protégée contre les attaques.
6. **Vérifier la synchronisation de la base de données** : Vous devez vous assurer que la base de données est synchronisée avec l'application Android et que les données sont correctement mises à jour. Vous pouvez utiliser des commandes SQL pour vérifier que les données sont correctement synchronisées entre l'application et la base de données.

En résumé, il est important de vérifier la disponibilité, les autorisations, la configuration, la performance, la sécurité et la synchronisation de la base de données lors du déploiement d'une application Android avec Amazon RDS et MySQL Workbench. Les vérifications doivent être effectuées régulièrement pour garantir que la base de données fonctionne correctement et que les données sont sécurisées et synchronisées entre l'application et la base de données.

5 - PROCÉDURE DE DÉMARRAGE / ARRÊT

5.1 - Avantages

- L'arrêt et le démarrage d'une instance de bases de données est plus rapide que la création et la restauration d'une instance de la base de données.
- Lorsque vous arrêtez une instance de bases de données, elle conserve ses ID, point de terminaison DNS (Domain Name Server), groupe de paramètres, groupe de sécurité et groupe d'options. Lorsque vous démarrez une instance de bases de données, sa configuration est la même que lorsque vous l'avez arrêtée. En outre, si vous arrêtez une instance de bases de données, Amazon RDS conserve les journaux de transactions Amazon S3. Cela signifie que vous pouvez effectuer une restauration à un point donné dans le passé, si nécessaire.

5.2 - Limites

Voici quelques limitations pour l'arrêt et le démarrage de votre instance de bases de données :

- Vous ne pouvez pas arrêter une instance de base de données qui possède un réplica en lecture ou qui est un réplica en lecture.
- Vous ne pouvez pas arrêter une instance de bases de données Amazon RDS for SQL Server dans une configuration multi-AZ.
- Vous ne pouvez pas modifier une instance de bases de données arrêtée.
- Vous ne pouvez pas supprimer un groupe d'options associé à une instance de bases de données arrêtée.
- Vous ne pouvez pas supprimer un groupe de paramètres de base de données associé à une instance de bases de données arrêtée.
- Dans une configuration multi-AZ, vous pouvez basculer entre les zones de disponibilité principale et secondaire après avoir démarré l'instance de base de données.

5.3 - Arrêt temporaire

- 1- Connectez-vous à la **AWS Management Console** et ouvrez la console Amazon RDS à l'adresse <https://console.aws.amazon.com/rds/>.
- 2- Dans le panneau de navigation, choisissez **Bases de données**, puis l'instance de base de données que vous voulez arrêter.
- 3- Pour Actions, choisissez **Stop temporarily (Arrêter temporairement)**.
- 4- Dans la fenêtre **Stop DB instance temporarily (Arrêter temporairement l'instance de base de données)**, sélectionnez l'accusé de réception indiquant que l'instance de base de données redémarrera automatiquement au bout de 7 jours.
- 5- (Facultatif) Sélectionnez **Save the DB instance in a snapshot (Enregistrer l'instance de base de données dans un instantané)** et saisissez le nom de l'instantané pour Snapshot name (Nom d'instantané). Choisissez cette option si vous souhaitez créer un instantané de l'instance de base de données avant de l'arrêter.
- 6- Choisissez **Stop temporarily (Arrêter temporairement)** pour arrêter l'instance de base de données ou choisissez **Cancel (Annuler)** pour annuler l'opération.

6 - PROCÉDURE DE MISE À JOUR

Voici une procédure de mise à jour de la base de données avec le logiciel MySQL Workbench :

1. **Ouvrir MySQL Workbench** : Lancez MySQL Workbench et connectez-vous à votre serveur de base de données.
2. **Sélectionner la base de données** : Sélectionnez la base de données que vous souhaitez mettre à jour dans le volet gauche de MySQL Workbench.
3. **Créer une sauvegarde de la base de données** : Avant de procéder à la mise à jour de la base de données, il est important de créer une sauvegarde de la base de données existante afin de restaurer la base de données en cas de problème lors de la mise à jour. Pour créer une sauvegarde, cliquez sur le menu "Serveur" en haut de MySQL Workbench, sélectionnez "Utilitaires de sauvegarde" et suivez les instructions.
4. **Modifier le modèle de base de données** : Modifiez le modèle de base de données pour ajouter ou supprimer des tables, des colonnes ou des contraintes. Vous pouvez utiliser l'éditeur de modèle de base de données dans MySQL Workbench pour effectuer ces modifications.
5. **Générer le script SQL** : Après avoir modifié le modèle de base de données, vous devez générer un script SQL qui exécutera les modifications. Cliquez sur le menu "Database" en haut de MySQL Workbench, sélectionnez "Forward Engineer" et suivez les instructions pour générer le script SQL.
6. **Exécuter le script SQL** : Exécutez le script SQL généré pour appliquer les modifications de la base de données. Cliquez sur le menu "Query" en haut de MySQL Workbench, collez le script SQL et exécutez-le.
7. **Vérifier la base de données** : Vérifiez que la base de données a été mise à jour avec succès. Vous pouvez utiliser l'éditeur de base de données dans MySQL Workbench pour vérifier que les modifications ont été appliquées.
8. **Tester l'application** : Après avoir mis à jour la base de données, vous devez tester votre application pour vous assurer que tout fonctionne correctement.

Il est important de prendre des précautions avant de mettre à jour une base de données, notamment en créant une sauvegarde de la base de données existante. Il est également important de tester l'application après la mise à jour pour garantir que tout fonctionne correctement.

7 - SUPERVISION/MONITORING

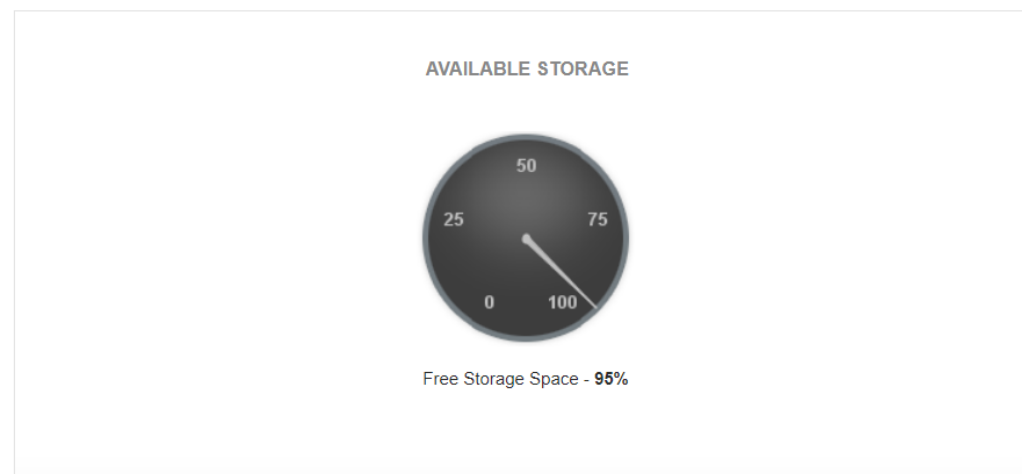
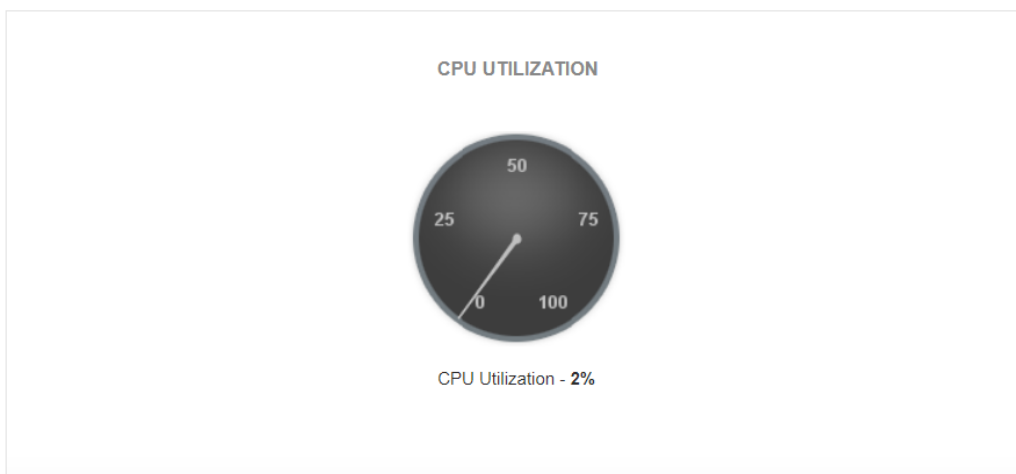
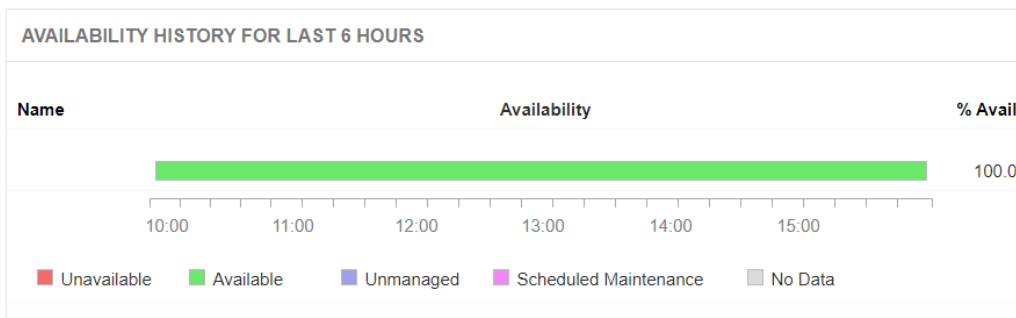
1. Amazon RDS est un service Web qui facilite la configuration, l'exploitation et la mise à l'échelle d'une base de données relationnelle dans le cloud. De nombreuses organisations choisissent d'héberger leurs bases de données de production critiques dans des instances Amazon RDS afin de profiter des avantages qu'elle offre. Il est donc essentiel de suivre ces instances avec un outil de surveillance d'Amazon RDS pour garantir que les services critiques pour l'entreprise ne sont pas compromis, et également pour s'assurer que les ressources cloud sont utilisées efficacement.
2. L'outil de supervision AWS d'Applications Manager de ManageEngine détecte automatiquement les instances Amazon RDS et surveille en permanence les indicateurs de performance clés de l'instance RDS. Avec des rapports prêts à l'emploi, des vues graphiques, des alarmes, des seuils et des capacités complètes de gestion des pannes, il offre une visibilité sur les performances et l'utilisation des ressources des instances RDS et des bases de données exécutées sur ces instances. Il permet également de vérifier les ruptures dans les accords de niveau de service et de garantir une expérience utilisateur de haute qualité.

Certaines des mesures fournies par le moniteur Amazon RDS d'Applications Manager comprennent :

- Disponibilité ;
- Utilisation du processeur ;
- Espace de stockage.



Supervisez minutieusement les performances RDS en suivant le nombre de connexions de base de données actives, les détails du débit, la latence de lecture/écriture du réseau et le nombre moyen d'opérations de lecture/écriture sur disque par seconde.



Monitor Information

Overview

Disk I/O

Network

Configuration

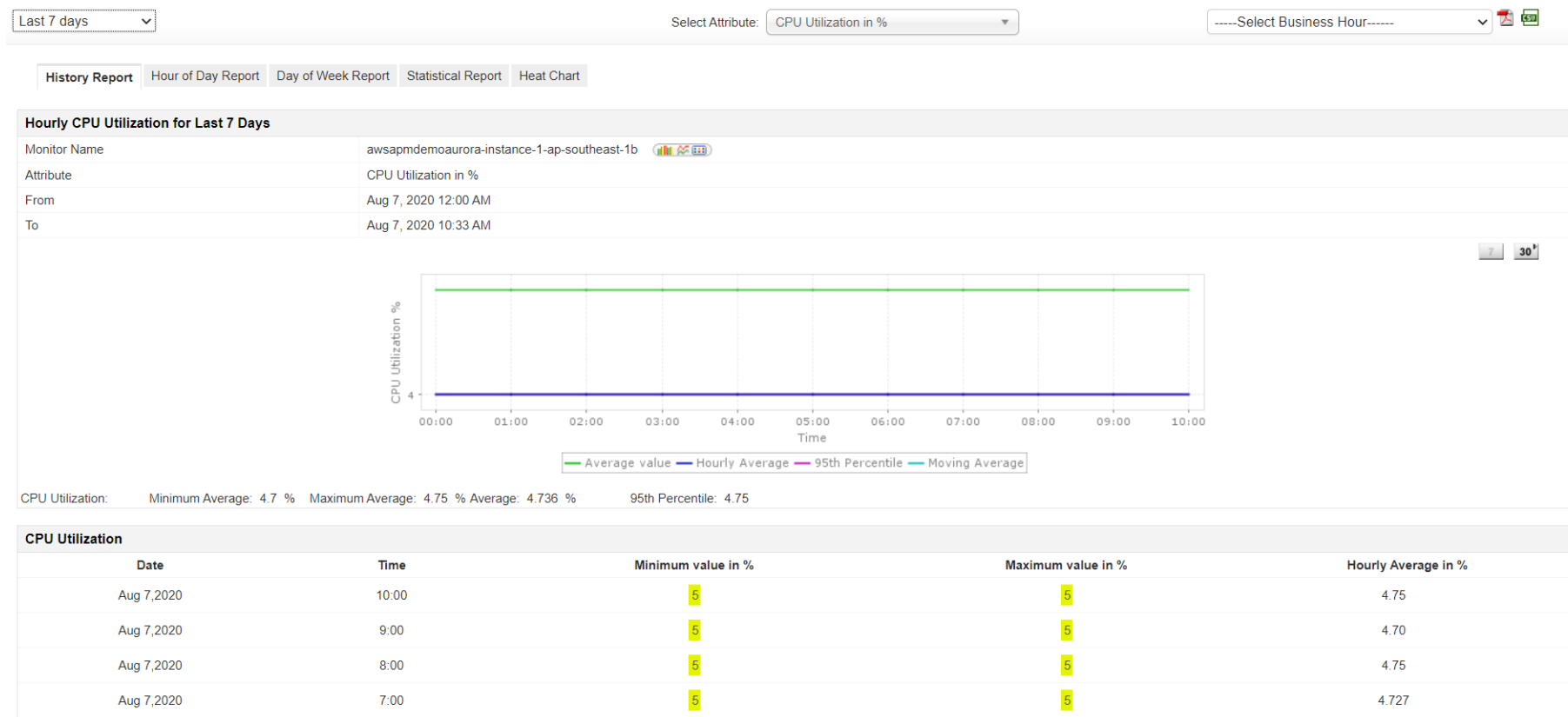
NETWORK TRAFFIC



Performance metrics from Aug 5,2020 12:00 AM to Aug 5,2020 09:46 AM with 45 minutes granularity

7.1.1 - Rapports de performance prêts à l'emploi

Applications Manager de ManageEngine pour la supervision des performances d'Amazon RDS fournit également des rapports prêts à l'emploi qui aident à analyser les performances globales des instances RDS et à planifier la capacité.



8 - PROCÉDURE DE SAUVEGARDE ET RESTAURATION

Amazon RDS offre deux options de sauvegarde et de restauration pour les instances de base de données :

- **les sauvegardes automatiques et les instantanés de base de données** : Les sauvegardes automatiques, activées par défaut, permettent une restauration à un instant précis grâce aux sauvegardes quotidiennes et aux journaux de transactions. La période de rétention maximale est de 35 jours.
- **les instantanés de base de données, initiés par l'utilisateur** : permettent de restaurer l'instance à un état spécifique à tout moment.

Les deux options sont compatibles avec AWS Backup, qui offre des fonctionnalités supplémentaires pour gérer les sauvegardes Amazon RDS.

9 - RESSOURCES

- Aide SQL. :
<https://sql.sh/>
- MySQL Workbench. :
<https://pandorafms.com/blog/fr/mysql-workbench/>
- Amazon RDS (12 mois gratuit) :
<https://aws.amazon.com/fr/rds/>
- Manageengine (30 jours offert) :
https://www.manageengine.com/fr/applications_manager/amazon-rds-monitoring.html
- Amazon Backup :
<https://aws.amazon.com/fr/backup-restore/>

10 - GLOSSAIRE

Client	Un client est un logiciel qui a pour but d'envoyer une demande à travers des requêtes afin de communiquer avec le serveur (l'interroger et attendre son retour).
Serveur	Un serveur est un dispositif informatique (matériel et logiciel) qui offre des services à un ou plusieurs clients. Il va répondre aux différentes requêtes émanant des clients.
Base de données	Une base de données est un « conteneur » stockant des données telles que des chiffres, des dates ou des mots, pouvant être retraités par des moyens informatiques pour produire une information. Par exemple, des chiffres et des noms assemblés et triés pour former un annuaire.
PostgreSQL	Il s'agit d'un système de gestion de bases de données relationnelles orienté objet et un logiciel gratuit qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes
MySQL	MySQL, de l'anglais Structured Query Language, est un système de gestion de bases de données relationnelles et un logiciel gratuit
Amazon Relational Database Service	C'est un service web qui facilite la configuration, l'exploitation et la mise à l'échelle d'une base de données relationnelle dans le Cloud. Il fournit des capacités redimensionnables, à faible coût, pour les bases de données relationnelles classiques, et gère les tâches courantes d'administration de base de données.
API	<p>Les interfaces de programmation d'application, de l'anglais Application Programming Interface (API) permettent à un produit ou service de communiquer avec d'autres produits et services sans connaître les détails de leur mise en œuvre. Elles offrent plus de flexibilité, simplifient la conception, l'administration et l'utilisation.</p> <p>Le Rest API (ou API Restfull) de l'anglais Representational State Transfer Application Program Interface, est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil.</p>
Material Design	Google a utilisé le Material Design pour unifier le style graphique de ses applications et de ses plateformes. Son interface présente donc l'avantage d'être très adaptée à une utilisation pour tout appareil et toutes résolutions d'affichage. Elle est également simple d'utilisation. Le Material Design optimiserait l'expérience utilisateur, tout en offrant des ressources aux développeurs et aux concepteurs d'interfaces.
Stripe	Stripe est une suite d'API offrant des solutions de traitement des paiements et de commerce en ligne pour les entreprises de toutes tailles.

Git	C'est un système de contrôle de version d'un projet et un logiciel gratuit. Il permet de traquer tous les fichiers d'un projet. On garde un historique de tous les changements et à tout moment nous pourrions revenir en arrière.
Android	C'est un système d'exploitation créé par Google qui permet la communication entre un utilisateur et son appareil. C'est un petit peu comme Windows, mais sur smartphone et tablettes.
Query	Query veut dire émettre une requête dans une base de données.
Backup	Backup veut dire sauvegarde. Cela désigne une opération de copier des fichiers ou données dans le but de protéger les données.
CPU	Le processeur de l'anglais Central Processing Unit (CPU) est un composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques.
DB	DB veut dire Base de données, de l'anglais Database.
CREATE	Symbolise la création. Ex. : CREATE TABLE (Créer TABLE).
SELECT	Symbolise la sélection. Ex. : SELECT TABLE (Sélectionner TABLE).
INSERT	Symbolise l'insertion. Ex. : INSERT TABLE (Insertion TABLE).
DROP	Symbolise la suppression. Ex. : DROP TABLE (Sélectionner TABLE).
ALTER	Symbolise la modification. Ex. : ALTER TABLE (Modifier TABLE).
Amazon RDS Multi-AZ	Amazon RDS Multi-AZ offre une disponibilité et une durabilité accrues pour les instances de base de données Amazon RDS, ce qui en fait une solution naturelle pour les charges de travail de base de données de production. Avec deux options de déploiement différentes, il est possible de personnaliser les charges de travail pour obtenir la disponibilité dont elles ont besoin.
Instance	Dans la modélisation UML, les instances de composant sont des éléments de modèle qui représentent des entités réelles dans un système.
MySQL Workbench	MySQL Workbench est un logiciel de gestion et d'administration de bases de données MySQL.