Compte rendu

Scheffer Benjamin

Unternehr David

Question 1 :

Sécurité : Quels sont les risques liés à l'utilisation de mots de passe en dur dans les fichiers de configuration ?

Les risques liés à l'utilisation de mots de passe en dur dans les fichiers de configuration sont :

-Risque de fuite : si le fichier fuite (sur un dépôt git ou autre) il sera très facile de trouver le mot de passe et donc un risque certain d'attaque de la base de données.

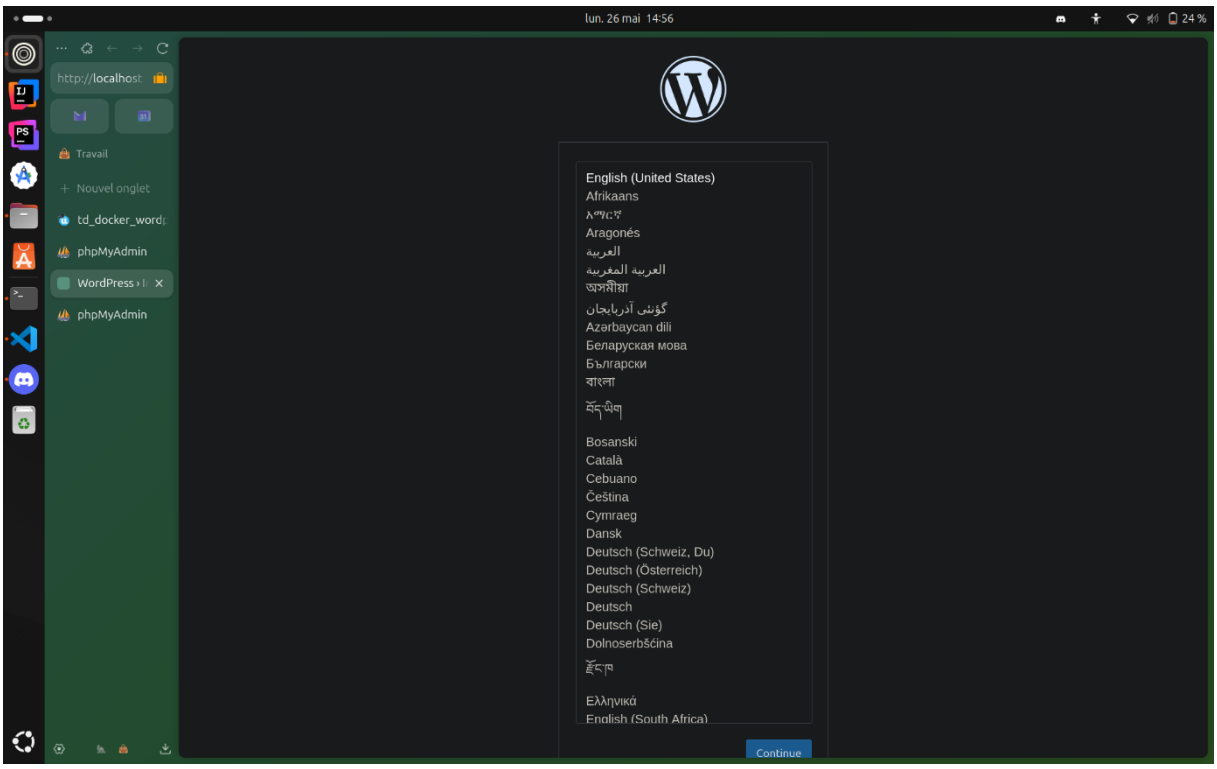
-Difficulté pour changer le mot de passe étant donné qu'il est écrit en dure dans le fichier de config il sera plus difficile de changer le mot de passe. Ainsi, il risque de ne pas être beaucoup changé donc plus de risque.

Question 2 :

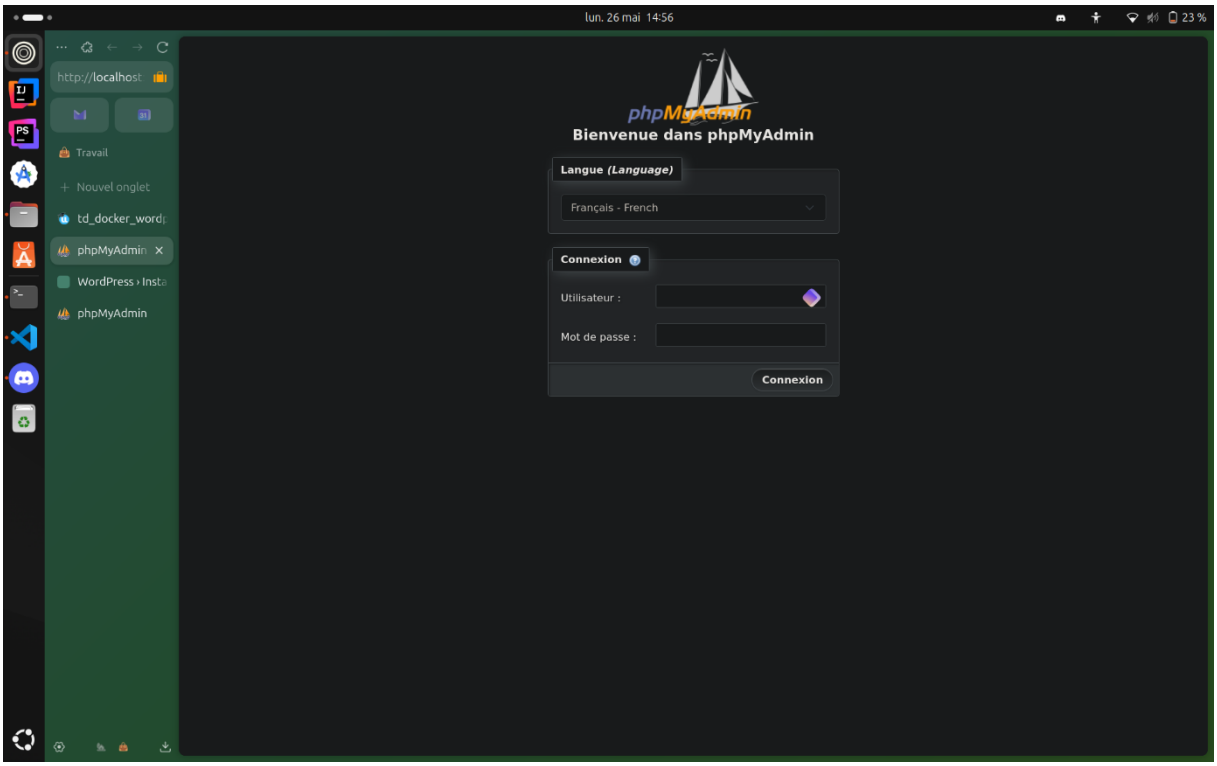
Réseau : Que signifie l'utilisation de '%' dans la création de l'utilisateur MySQL ?

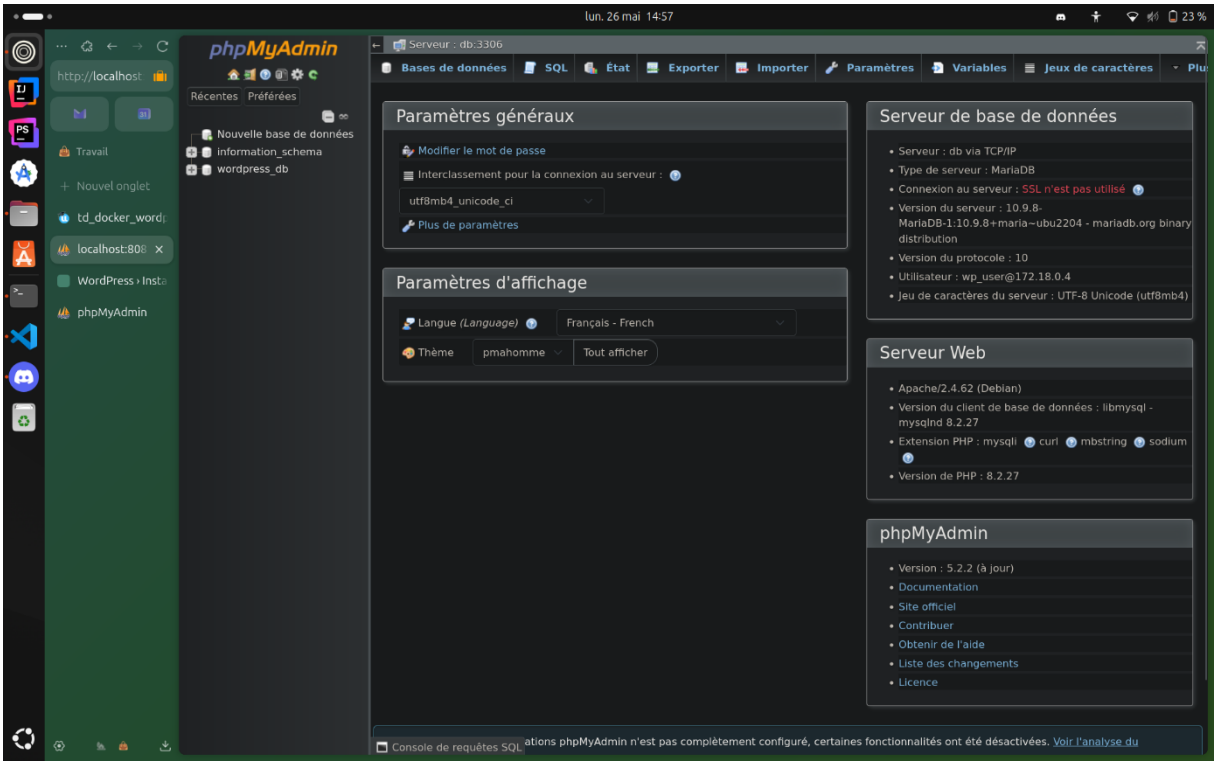
L'utilisation de '%' dans la création de l'utilisateur MySQL signifie que toutes les ip pourront accéder à ce compte.

Accès WordPress :



Accès PHPMyAdmin :





Exercices pratiques :

Exercice 1 : Analyse des logs :

Logs maria DB :

```
david@david-MCLF-XX:~/Documents/GitHub/Virtualisation-S4/wordpress-stack$ sudo docker-compose logs db
WARN[0000] /home/david/Documents/GitHub/Virtualisation-S4/wordpress-stack/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pl
ease remove it to avoid potential confusion
wordpress_db  | 2025-05-26 13:02:59+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
wordpress_db  | 2025-05-26 13:02:59+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
wordpress_db  | 2025-05-26 13:02:59+00:00 [Note] [Entrypoint]: Entrypoint script for MariaDB Server 1:10.9.8+maria~ubu2204 started.
wordpress_db  | 2025-05-26 13:02:59+00:00 [Note] [Entrypoint]: MariaDB upgrade not required
wordpress_db  | 2025-05-26 13:02:59 0 [Note] Starting MariaDB 10.9.8-MariaDB-1:10.9.8+maria~ubu2204 source revision 3e0009dc3a771e4dbf2fa4a4cf87e750453fb
2eb as process 1
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Compressed tables use zlib 1.2.11
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Number of transaction pools: 1
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Using crc32 + pclmulqdq instructions
wordpress_db  | 2025-05-26 13:02:59 0 [Warning] mariadbd: io_uring_queue_init() failed with errno 1
wordpress_db  | 2025-05-26 13:02:59 0 [Warning] InnoDB: liburing disabled: falling back to innodb_use_native_aio=OFF
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Initializing buffer pool, total size = 128.000MiB, chunk size = 2.000MiB
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Completed initialization of buffer pool
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: File system buffers for log disabled (block size=512 bytes)
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: End of log at LSN=47092
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: 128 rollback segments are active.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Setting file './ibtmp1' size to 12.000MiB. Physically writing the file full; Please wait ...
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: File './ibtmp1' size is now 12.000MiB.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: log sequence number 47092; transaction id 16
wordpress_db  | 2025-05-26 13:02:59 0 [Note] Plugin 'FEEDBACK' is disabled.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
wordpress_db  | 2025-05-26 13:02:59 0 [Warning] You need to use --log-bin to make --expire-logs-days or --binlog-expire-logs-seconds work.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] InnoDB: Buffer pool(s) load completed at 250526 13:02:59
wordpress_db  | 2025-05-26 13:02:59 0 [Note] Server socket created on IP: '0.0.0.0'.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] Server socket created on IP: '::'.
wordpress_db  | 2025-05-26 13:02:59 0 [Note] mariadbd: ready for connections.
wordpress_db  | Version: '10.9.8-MariaDB-1:10.9.8+maria~ubu2204'  socket: '/run/mysqld/mysqld.sock'  port: 3306  mariadb.org binary distribution
david@david-MCLF-XX:~/Documents/GitHub/Virtualisation-S4/wordpress-stack$
```

Logs WordPress :

```
david@david-MCLF-XX:~/Documents/GitHub/Virtualisation-S4/wordpress-stack$ sudo docker-compose logs wordpress
WARN[0000] /home/david/Documents/GitHub/Virtualisation-S4/wordpress-stack/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pl
ease remove it to avoid potential confusion
wordpress_app  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.5. Set the 'ServerName' directiv
e globally to suppress this message
wordpress_app  | AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.18.0.5. Set the 'ServerName' directiv
e globally to suppress this message
wordpress_app  | [Mon May 26 13:02:59.534989 2025] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.57 (Debian) PHP/8.1.25 configured -- resuming normal
operations
wordpress_app  | [Mon May 26 13:02:59.535011 2025] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'
```

Logs de redis :

```
david@david-MCLF-XX:~/Documents/GitHub/Virtualisation-S4/wordpress-stack$ sudo docker-compose logs redis
WARN[0000] /home/david/Documents/GitHub/Virtualisation-S4/wordpress-stack/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pl
ease remove it to avoid potential confusion
wordpress_redis  | 1:C 26 May 2025 13:02:59.321 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail under
low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To f
ix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effe
ct.
wordpress_redis  | 1:C 26 May 2025 13:02:59.321 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
wordpress_redis  | 1:C 26 May 2025 13:02:59.321 * Redis version=7.4.3, bits=64, commit=00000000, modified=0, pid=1, just started
wordpress_redis  | 1:C 26 May 2025 13:02:59.321 * Configuration loaded
wordpress_redis  | 1:M 26 May 2025 13:02:59.321 * Increased maximum number of open files to 10032 (it was originally set to 1024).
wordpress_redis  | 1:M 26 May 2025 13:02:59.321 * monotonic clock: POSIX clock_gettime
wordpress_redis  | 1:M 26 May 2025 13:02:59.322 * Running mode=standalone, port=6379.
wordpress_redis  | 1:M 26 May 2025 13:02:59.322 * Server initialized
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * Reading RDB base file on AOF loading...
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * Loading RDB produced by version 7.4.3
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * RDB age 1150 seconds
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * RDB memory usage when created 0.90 Mb
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * RDB is base AOF
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * Done loading RDB, keys loaded: 0, keys expired: 0.
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * DB loaded from base file appendonly.aof.1.base.rdb: 0.000 seconds
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * DB loaded from append only file: 0.000 seconds
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * Opening AOF incr file appendonly.aof.1.incr.aof on server start
wordpress_redis  | 1:M 26 May 2025 13:02:59.323 * Ready to accept connections tcp
```

Question : Que se passe-t-il si WordPress démarre avant que MariaDB soit complètement initialisée ?

Si WordPress démarre avant MariaDB, WordPress ne pourra pas se connecter à la base de données.

Exercice 2 : Tests de connectivité :

Avant de réaliser les différentes commandes nous avons dû ajouter et modifier plusieurs composants du DOCKERFILE. Pourquoi ?
Dans un premier temps il manquait l'installation des commandes mysql et redis-cli

donc afin d'éviter d'installer les commandes directement dans le container nous les avons ajoutés au dockerfile. Ainsi lorsqu'on relance le container les commandes sont déjà là on évite donc de les réinstaller à chaque coup.

Enfin l'url du curl de la dernière partie du fichier DOCKERFILE afficher une erreur 404. Cela arrive quand on dépend d'une image généralement non officielle. L'auteur de cette image peu avoir changer le nom de l'url ou autre ainsi nous n'avons plus accès à cette image. Nous avons donc remplacé cette url par une url officiel de WordPress.

Image du DOCKERFILE :

```
wordpress-stack > wordpress > Dockerfile > ...
1  FROM wordpress:6.3-php8.1-apache
2
3  # Installation des extensions PHP nécessaires
4  RUN apt-get update && apt-get install -y \
5      default-mysql-client \
6      redis-tools \
7      libzip-dev \
8      zip \
9      unzip \
10     && docker-php-ext-install zip
11
12 # Installation de l'extension Redis pour PHP
13 RUN pecl install redis \
14     && docker-php-ext-enable redis
15
16 # Configuration personnalisée de PHP
17 COPY php.ini /usr/local/etc/php/conf.d/custom.ini
18
19 # Installation de WP-CLI
20 RUN curl -O https://github.com/wp-cli/wp-cli/releases/download/v2.12.0/wp-cli-2.12.0.phar \
21     && chmod +x wp-cli-2.12.0.phar \
22     && mv wp-cli-2.12.0.phar /usr/local/bin/wp
```

Les lignes ajoutés et modifiés sont ligne 5,6 et 20.

Résultat des commandes :

```
root@f7076a4423a5:/var/www/html# mysql -h db -u wp_user -p wordpress_db
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.9.8-MariaDB-1:10.9.8+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
root@f7076a4423a5:/var/www/html# redis-cli -h redis ping
PONG
```

Exercice 3 : Sauvegarde et restauration

Ici la commande avec root ne fonctionner pas nous avons donc changer root par wp_user en utilisant le mot de passe wp_password.

Résultat du backup :

wordpress-stack > backup.sql

```
-- MariaDB dump 10.19  Distrib 10.9.8-MariaDB, for debian-linux-gnu (x86_64)
--
-- Host: localhost    Database: wordpress_db
-- ------------------------------------------------------
-- Server version   10.9.8-MariaDB-1:10.9.8+maria~ubu2204

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2025-05-26 14:26:03
```

Pour restaurer la base de donnée on utiliserait dans le conteneur db la commande : mysql -u wp_user -p wordpress_db < backup.sql

Exercice 4 : Monitoring des ressources

Résultat de la commande :

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
f4a18a0f7578	wordpress_app	0.00%	14.27MiB / 15.35GiB	0.09%	8.37kB / 0B	0B / 8.19kB	6
6407bc0ebf6d	wordpress_phpmyadmin	0.01%	12.09MiB / 15.35GiB	0.08%	8.37kB / 0B	0B / 8.19kB	6
8a0391e3b539	wordpress_redis	0.92%	3.273MiB / 15.35GiB	0.02%	8.71kB / 0B	0B / 0B	6
15a835d6ed38	wordpress_db	0.02%	76.22MiB / 15.35GiB	0.48%	8.71kB / 0B	4.7MB / 8.19kB	8

Question :

Quel conteneur consomme le plus de mémoire ?

Donc le container qui utilise le plus de mémoire est le container de la base de données

Comment pourriez-vous limiter l'utilisation des ressources ?

Il y a plusieurs options pour limiter l'utilisation des ressources :

- Réaliser des DOCKERFILE pour limiter le nombre d'installation
- Utiliser des images moins volumineuses.
- Limiter l'utilisation des composants du PC (Ram,CPU ...)

Questions de réflexion et d'approfondissement

Architecture et conception

1. **Expliquez l'avantage d'utiliser des conteneurs séparés pour chaque service plutôt qu'un seul conteneur avec tous les services.**

Cela permet d'isoler les services. L'isolation va permettre de pouvoir réaliser des mises a niveau de chaque service selon nos besoins sans impacter les autres services. On pourra aussi plus facilement réutiliser les services quand ils sont isolés que dans un seul container.

2. **Quels sont les avantages et inconvénients du stockage des données dans des volumes Docker par rapport au stockage dans le conteneur ?**

Les avantages :

- Cela permet d'avoir de la persistance dans les données. C'est-à-dire au moment ou nous allons couper le container il va falloir sauvegarder les données. Or avec les volumes pas besoin de sauvegarder à chaque coup.

- Mettre à jour les conteneurs sans perdre les données.

- Les volumes sont plus rapide que les fichiers dans un conteneur.

- Possibilité de partager des données entre plusieurs conteneurs

Les inconvénients :

- Demande plus de configuration.

3. **Comment le réseau Docker permet-il la communication entre conteneurs ?**

Il y a plusieurs outils pour réaliser des communications entre conteneurs :

-Les bridges. Un bridge est un système qui permet aux conteneurs qui y sont connectés de communiquer entre eux. (SRC : cours « Introduction a Docker »).

-Les hosts. Un host est une instance d'un conteneur dans le réseau.

Dans un fichier docker-compose.yml, on utilise la propriété depends_on : [nom du container] afin de créer un bridge entre le container actuel et web.

Sécurité

1. **Identifiez trois vulnérabilités potentielles dans cette configuration et proposez des solutions.**

1 : Les mots de passes stockées en claire dans les fichiers de configuration (ex : docker-compose.yml ou encore init.sql)

Solution :

Utiliser des variables d'environnement. Pour cela on remplace dans le docker-compose.yml les endroits des mots de passes, des hôtes et autres par :

${Le_Nom_Attribuer}

2 : Il y a une exposition directe des services sur des ports publics comme PhPMyAdmin : avec les ports : 8081.

Solution :

Rajouté une image Nginx afin d'ajouté un proxy inverse.

3 : Manque de sécurisation des données. En effet en cas d'erreur ou autre de la base de données MariaDB nous n'avons aucun backup.

Solution :

Faire un script de sauvegarde des données notamment avec la commande : mysqldump -u ... -p ... > backup.sql

2. **Comment pourriez-vous chiffrer les communications entre les conteneurs ?**

On pourrait par exemple créer un réseau overlay (donc pour tout les conteneurs) et crypter les communications sur ce réseaux grâce à Docker Swarm.

Performance et scalabilité

1. **Comment Redis améliore-t-il les performances de WordPress ?**

Redis améliore les performances de WordPress notamment en stockant en mémoire la Session et en mettant en cache des données ce qui évite un certain nombre de requête SQL.

2. **Proposez une stratégie pour gérer une montée en charge (scaling horizontal).**

Pour gérer une montée en charge il faut utiliser un load balencer qui aura pour but de créer conteneur selon les montées en charge. On peut pour cela utiliser Kubernetes en configurant un Horizontal Pod Autoscaler. Et ici c'est le conteneur wordpress qu'on va dupliquer.

DevOps et production

1 **Quelles modifications apporteriez-vous pour un déploiement en production ?**

Pour réaliser un déploiement en production je comblerais tous les défauts cités dans l'une des premières questions. C'est-à-dire mettre en place un proxy, mettre les mots de passes et autres dans des variables d'environnement et ajouté un système de sauvegarde automatique.

2 **Comment intégreriez-vous ce stack dans un pipeline CI/CD ?**

Cela va dépendre du registre qu'on souhaite utiliser mais par exemple avec un GitLab on va devoir créer un fichier .gitlab-ci.yml qui contient tout le nécessaire pour build les images, les tests et les déployer.