



Création

Résumé : Ce document est un exercice relatif à l'administration système.

Version : 4.0

Contenu

je	Préambule	2
II	Introduction	3
III	Règles générales	4
IV	Instructions d'IA	5
V	Partie obligatoire	7
VI	Partie bonus	12
VII	Soumission et évaluation par les pairs	13

Chapitre I

Préambule



Chapitre II

Introduction

Ce projet vise à approfondir vos connaissances en administration système grâce à l'utilisation de la technologie Docker. Vous virtualiserez plusieurs images Docker en les créant dans votre nouvelle machine virtuelle personnelle.

Chapitre III

Règles générales

- Ce projet doit être réalisé sur une machine virtuelle.
- Tous les fichiers nécessaires à la configuration de votre projet doivent être placés dans un dossier.
- `UNMakefile` Ce fichier est également requis et doit se trouver à la racine de votre répertoire. Il doit configurer l'intégralité de votre application (c'est-à-dire qu'il doit créer les images Docker). `docker-compose.yml`).
- Ce sujet requiert la mise en pratique de concepts que vous n'avez peut-être pas encore acquis, selon votre niveau de formation. Par conséquent, nous vous conseillons de consulter la documentation détaillée relative à l'utilisation de Docker, ainsi que toute autre ressource que vous jugerez utile pour réaliser ce travail.

Chapitre IV

Instructions d'IA

● Contexte

Tout au long de votre apprentissage, l'IA peut vous assister dans de nombreuses tâches. Prenez le temps d'explorer les différentes fonctionnalités des outils d'IA et comment ils peuvent vous aider dans votre travail. Toutefois, abordez-les toujours avec prudence et évaluez les résultats de manière critique. Qu'il s'agisse de code, de documentation, d'idées ou d'explications techniques, vous ne pouvez jamais être absolument certain que votre question était bien formulée ni que le contenu généré est exact. Vos pairs constituent une ressource précieuse pour vous aider à éviter les erreurs et les angles morts.

● Message principal

- 👉 Utilisez l'IA pour réduire les tâches répétitives ou fastidieuses.
- 👉 Développez des compétences en matière de stimulation émotionnelle — avec ou sans codage — qui vous seront utiles pour votre future carrière.
- 👉 Découvrez comment fonctionnent les systèmes d'IA pour mieux anticiper et éviter les risques, les biais et les problèmes éthiques courants.
- 👉 Continuez à développer vos compétences techniques et relationnelles en travaillant avec vos pairs.
- 👉 N'utilisez que du contenu généré par l'IA que vous comprenez parfaitement et dont vous pouvez assumer la responsabilité.

● Règles de l'apprenant :

- Vous devriez prendre le temps d'explorer les outils d'IA et de comprendre leur fonctionnement, afin de pouvoir les utiliser de manière éthique et réduire les biais potentiels.
- Il est conseillé de bien réfléchir à votre problème avant de formuler des suggestions ; cela vous permettra de rédiger des suggestions plus claires, plus détaillées et plus pertinentes, en utilisant un vocabulaire précis.
- Vous devriez prendre l'habitude de vérifier, d'examiner, de questionner et de tester systématiquement tout ce qui est généré par l'IA.
- Vous devriez toujours solliciter une évaluation par les pairs — ne vous fiez pas uniquement à votre propre validation.

● Résultats de la phase :

- Développer des compétences d'incitation à la fois générales et spécifiques au domaine.
- Boostez votre productivité grâce à une utilisation efficace des outils d'IA.
- Continuez à développer la pensée computationnelle, la résolution de problèmes, l'adaptabilité et la collaboration.

● Commentaires et exemples :

- Vous serez régulièrement confronté à des situations — examens, évaluations, etc. — où vous devrez démontrer une réelle compréhension. Soyez prêt et continuez à développer vos compétences techniques et relationnelles.
- Expliquer son raisonnement et débattre avec ses pairs permet souvent de déceler les lacunes de sa compréhension. Il est donc essentiel de privilégier l'apprentissage entre pairs.
- Les outils d'IA manquent souvent de contexte spécifique et tendent à fournir des réponses génériques. Vos pairs, qui partagent votre environnement, peuvent vous apporter des informations plus pertinentes et précises.
- Là où l'IA tend à fournir la réponse la plus probable, vos collègues peuvent apporter des perspectives alternatives et des nuances précieuses. Fiez-vous à eux pour un contrôle qualité.

✓ Bonnes pratiques :

Je demande à l'IA : « Comment tester une fonction de tri ? » Elle me suggère quelques pistes. Je les teste et j'en discute avec un collègue. Nous affinons ensuite la méthode ensemble.

✗ Mauvaise pratique :

Je demande à une IA d'écrire une fonction complète, puis je la copie-colle dans mon projet. Lors de l'évaluation par les pairs, je suis incapable d'expliquer son fonctionnement ni sa raison d'être. Je perds toute crédibilité et mon projet échoue.

✓ Bonnes pratiques :

J'utilise l'IA pour concevoir un analyseur syntaxique. Ensuite, je passe en revue la logique avec un collègue. Nous repérons deux bogues et réécrivons le code ensemble : il est meilleur, plus clair et parfaitement compréhensible.

✗ Mauvaise pratique :

J'ai laissé Copilot générer le code d'une partie essentielle de mon projet. La compilation réussit, mais je ne parviens pas à expliquer comment il gère les pipes. Lors de l'évaluation, je ne peux pas justifier mon erreur et mon projet échoue.

Chapitre V

Partie obligatoire

Ce projet consiste à mettre en place une petite infrastructure composée de différents services fonctionnant selon des règles spécifiques. L'ensemble du projet doit être réalisé dans une machine virtuelle. L'utilisation de Docker Compose est obligatoire.

Chaque image Docker doit porter le même nom que le service correspondant. Chaque service doit s'exécuter dans un conteneur dédié.

Pour des raisons de performance, les conteneurs doivent être créés à partir de l'avant-dernière version stable d'Alpine ou de Debian. Le choix vous appartient.

Vous devez également écrire le vôtre Dockerfiles, un par service. Dockerfiles doit être appelé dans votre docker-compose.yml par votre Makefile.

Cela signifie que vous devez créer vous-même les images Docker de votre projet. Il est donc interdit d'utiliser des images Docker préexistantes ou des services comme Docker Hub (Alpine/Debian faisant exception à cette règle).

Vous devez ensuite configurer :

- Un conteneur Docker contenant NGINX avec TLSv1.2 ou TLSv1.3 uniquement.
- Un conteneur Docker contenant WordPress avec php-fpm (qui doit être installé et configuré), sans nginx.
- Un conteneur Docker contenant uniquement MariaDB, sans nginx.
- Un volume qui contient votre base de données WordPress.
- Un deuxième volume contenant les fichiers de votre site web WordPress.
- UN réseau Docker qui établit la connexion entre vos conteneurs.

Vos conteneurs doivent redémarrer automatiquement en cas de panne.



Un conteneur Docker n'est pas une machine virtuelle. Par conséquent, il est déconseillé d'utiliser des solutions de contournement non conventionnelles basées sur la commande « tail -f » ou des méthodes similaires pour l'exécuter. Renseignez-vous sur le fonctionnement des démons et déterminez s'il est judicieux de les utiliser.



Bien entendu, l'utilisation de `network: host`, `--link` ou `links:` est interdite. La ligne `network` doit impérativement figurer dans votre fichier `docker-compose.yml`. Vos conteneurs ne doivent pas être lancés par une commande exécutant une boucle infinie. Par conséquent, cette interdiction s'applique également à toute commande utilisée comme point d'entrée ou dans les scripts de point d'entrée. Voici quelques exemples de solutions de contournement interdites : `tail -f`, `bash`, `sleep infinity`, `while true`.



Lisez la documentation sur le PID 1 et les bonnes pratiques pour la rédaction des Dockerfiles.

- Votre base de données WordPress doit comporter deux utilisateurs, dont l'administrateur. Le nom d'utilisateur de l'administrateur ne doit pas contenir « admin », « Admin », « administrator » ou « Administrator » (par exemple : admin, administrator, Administrator, admin-123, etc.).



Vos volumes seront disponibles dans le dossier `/home/login/data` de la machine hôte via Docker. Bien entendu, vous devrez remplacer « login » par votre identifiant.

Pour simplifier le processus, vous devez configurer votre nom de domaine pour qu'il pointe vers votre adresse IP locale.

Ce nom de domaine doit être `connexion.42.fr`. Là encore, vous devez utiliser vos propres identifiants.

Par exemple, si votre identifiant est « wil », `wil.42.fr`. Vous serez redirigé vers l'adresse IP pointant vers le site web de Wil.



Le dernier tag est interdit.

Les mots de passe ne doivent pas figurer dans vos fichiers Dockerfile.

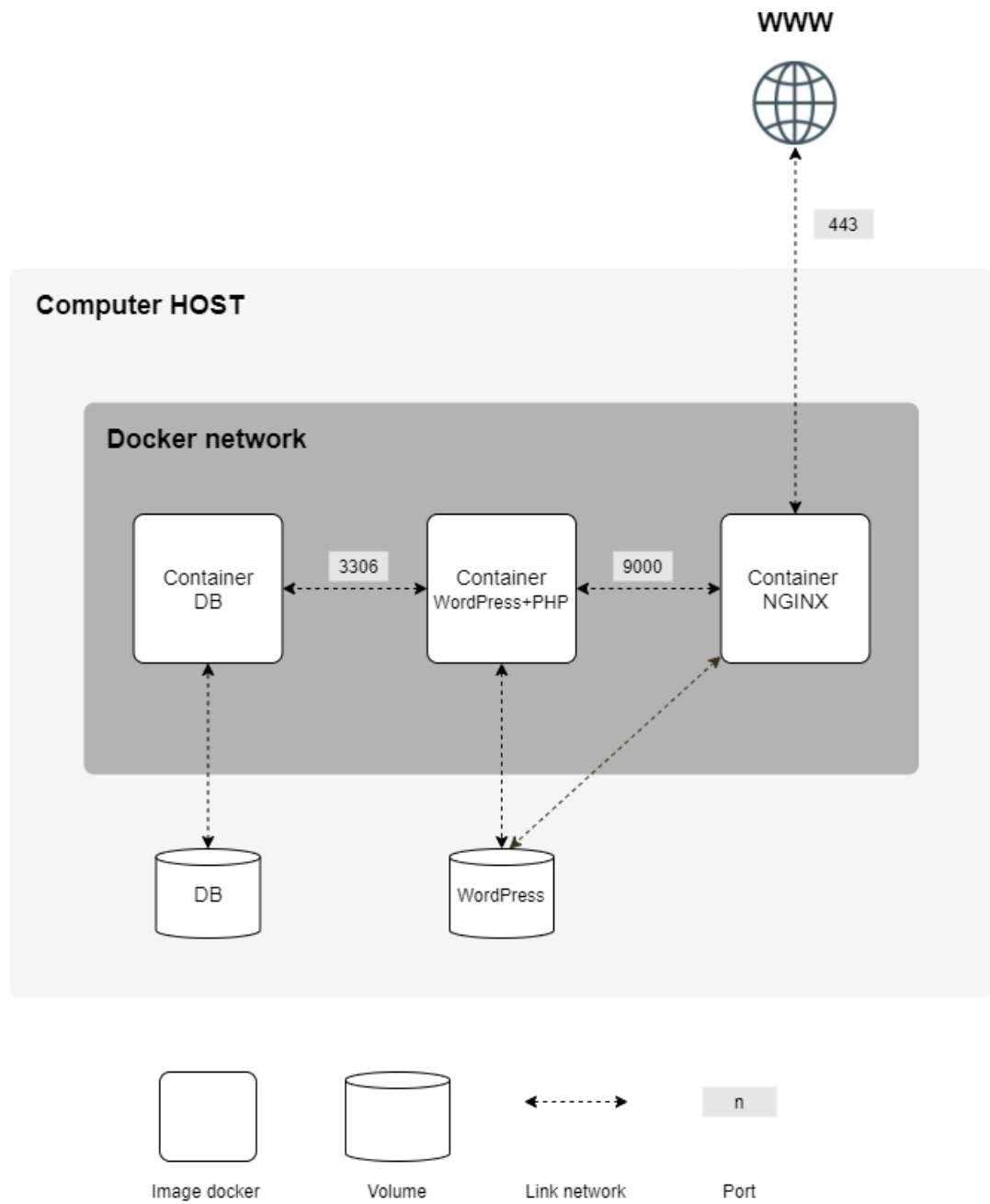
L'utilisation de variables d'environnement est obligatoire.

Il est également fortement recommandé d'utiliser un fichier `.env` pour stocker les variables d'environnement et d'utiliser les secrets Docker pour stocker toute information confidentielle.

Votre conteneur NGINX doit être le seul point d'entrée dans votre infrastructure, accessible uniquement via le port 443, utilisant le protocole TLSv1.2 ou TLSv1.3.

Création

Voici un exemple de diagramme illustrant le résultat attendu :



Voici un exemple de la structure de répertoires attendue :

```
$> ls -alR
total XX
drwxrwxr-x 3 sera le 4096 avril 42 20:42 . drwxrwxrwt
17 wil 4096 avril 42 20:42 ..
- rw-rw-r-- 1 wil wil XXXX 42 avril 20:42 Makefile drwxrwxr-x
3 wil wil 4096 42 avril 20:42 secrets
drwxrwxr-x 3 wil 4096 avril 42 20:42 srcs

./secrets:
total XX
drwxrwxr-x 2 sera le 4096 avril 42 20:42 .
drwxrwxr-x 6 wil 4096 avril 42 20:42 ..
- rw-r--r-- 1 wil wil XXXX avril 42 20:42 credentials.txt
- rw-r--r-- 1 wil wil XXXX avril 42 20:42 db_password.txt
- rw-r--r-- 1 wil wil XXXX avril 42 20:42 db_root_password.txt

./srcs:
total XX
drwxrwxr-x 3 sera le 4096 avril 42 20:42 .
drwxrwxr-x 3 wil 4096 avril 42 20:42 ..
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 docker-compose.yml
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .env
drwxrwxr-x 5 wil 4096 avril 42 20:42 exigences

./srcs/requirements :
total XX
drwxrwxr-x 5 sera 4096 avril 42 20:42 .
drwxrwxr-x 3 wil 4096 avril 42 20:42 ..
drwxrwxr-x 4 wil 4096 avril 42 20:42 bonus
drwxrwxr-x 4 wil 4096 avril 42 20:42 mariadb
drwxrwxr-x 4 wil 4096 avril 42 20:42 nginx
drwxrwxr-x 4 wil 4096 avril 42 20:42 outils
drwxrwxr-x 4 wil 4096 avril 42 20:42 wordpress

./srcs/requirements/mariadb :
total XX
drwxrwxr-x 4 sera 4096 avril 42 20:45 .
drwxrwxr-x 5 wil 4096 avril 42 20:42 ..
drwxrwxr-x 2 wil 4096 avril 42 20:42 conf
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil 4096 avril 42 20:42 outils
[...]

./srcs/requirements/nginx :
total XX
drwxrwxr-x 4 sera 4096 avril 42 20:42 .
drwxrwxr-x 5 wil 4096 avril 42 20:42 ..
drwxrwxr-x 2 wil 4096 avril 42 20:42 conf
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 Dockerfile
- rw-rw-r-- 1 wil wil XXXX avril 42 20:42 .dockerignore
drwxrwxr-x 2 wil 4096 avril 42 20:42 outils
[...]

$> cat srcs/.env
NOM_DE_DOMAINE=wil.42.fr
# CONFIGURATION MYSQL
MYSQL_USER=XXXXXXXXXXXX
[...]
$>
```



Pour des raisons évidentes de sécurité, tous les identifiants, clés API, mots de passe, etc., doivent être enregistrés localement dans des fichiers spécifiques et ignorés par Git. Le stockage public de ces identifiants entraînera inévitablement l'échec du projet.



Vous pouvez stocker vos variables (sous forme de nom de domaine) dans un fichier de variables d'environnement comme `.env`

Chapitre VI

Partie bonus

Pour ce projet, la partie bonus se veut simple.

Un fichier Dockerfile doit être créé pour chaque service supplémentaire. Ainsi, chaque service s'exécutera dans son propre conteneur et disposera, le cas échéant, de son propre volume dédié.

Liste bonus :

- Installation d'un cache Redis pour votre site WordPress afin de gérer correctement le cache.
- Configurer un serveur FTP conteneur pointant vers le volume de votre site web WordPress.
- Créez un site web statique simple dans le langage de votre choix, à l'exception de PHP (oui, PHP est exclu). Par exemple, un site vitrine ou un site pour présenter votre CV.
- Installation d'un administrateur.
- Mettez en place un service de votre choix que vous jugez utile. Lors de la soutenance, vous devrez justifier votre choix.



Pour compléter la partie bonus, vous avez la possibilité de configurer des services supplémentaires. Dans ce cas, vous pouvez ouvrir davantage de ports selon vos besoins.



La partie bonus ne sera évaluée que si la partie obligatoire est parfaitement réalisée. Par « parfaitement réalisée », on entend que la partie obligatoire a été entièrement complétée et fonctionne sans aucun dysfonctionnement. Si vous avez Si vous ne remplissez pas TOUTES les conditions obligatoires, votre prime ne sera pas versée. être évalués tout court.

Chapitre VII

Soumission et évaluation par les pairs

Soumettez votre devoir à votreGitLe dépôt reste inchangé. Seul le contenu de votre dépôt sera évalué lors de la soutenance. N'hésitez pas à revérifier l'exactitude des noms de vos dossiers et fichiers.

Au cours de l'évaluation, un bref**modification du projet**Une demande ponctuelle peut être formulée. Il peut s'agir d'une modification mineure du comportement, de quelques lignes de code à écrire ou à réécrire, ou d'une fonctionnalité facile à ajouter.

Bien que cette étape puisse**s'applique pas à tous les projets**, vous devez vous y préparer si cela est mentionné dans les directives d'évaluation.

Cette étape vise à vérifier votre compréhension d'une partie spécifique du projet. La modification peut être effectuée dans l'environnement de développement de votre choix (par exemple, votre environnement habituel) et devrait être réalisable en quelques minutes, sauf si un délai précis est défini dans le cadre de l'évaluation.

Il peut vous être demandé, par exemple, d'effectuer une petite mise à jour d'une fonction ou d'un script, de modifier un affichage ou d'ajuster une structure de données pour stocker de nouvelles informations, etc.

Les détails (portée, cible, etc.) seront précisés dans le**lignes directrices d'évaluation**et peuvent varier d'une évaluation à l'autre pour un même projet.



16D85ACC441674FBA2DF65190663EC3C3C258FEA065D090A715F1B62F5A57F0B75403
61668BD6823E2F873124B7E59B5CE94BB7ABD71CD01F65B959E14A3838E414F1E871
F7D91730B