**Exercice 1 – Remise en ordre et compréhension du cycle DevOps (3 points)**

Une entreprise souhaite mettre en place une organisation DevOps pour améliorer la livraison de ses applications.

Les activités suivantes sont proposées dans le désordre :

- A. Déploiement de l'application sur un environnement de production.
- B. Récupération du code source depuis Git.
- C. Exécution des tests automatiques.
- D. Compilation de l'application.
- E. Préparation de la version à distribuer.
- F. Collaboration entre les équipes de développement et d'exploitation.

Questions :

1. Classer les activités B, C, D et E dans un ordre logique pour construire une chaîne d'intégration continue. (1 pt)
2. À quel moment intervient le déploiement de l'application ? Expliquer. (0,5 pt)
3. Expliquer pourquoi la collaboration entre les développeurs et les équipes d'exploitation est essentielle dans une démarche DevOps. (1 pt)
4. Citer un bénéfice attendu de cette organisation. (0,5 pt)

**Exercice 2 – Analyse et correction de commandes Git (4 points)**

Un développeur travaille sur un projet nommé gestion-stock.

Il exécute les commandes suivantes :

```
git init
git status
git add src
git commit -m "Premier commit"
git remote add origin https://github.com/equipe/gestion-stock.git
git push origin master
```

Questions :

1. Expliquer le rôle des commandes git init, git add et git commit. (1,5 pt)
2. Quelle est la différence entre git add et git commit ? (0,5 pt)
3. Le développeur souhaite récupérer le projet sur une deuxième machine. Quelle commande peut-il utiliser pour copier le dépôt distant en local ? (0,5 pt)
4. Un autre développeur a effectué des modifications sur GitHub. Quelle commande permet de récupérer ces modifications sur le dépôt local ? (0,5 pt)
5. Le développeur souhaite consulter l'historique des modifications. Quelle commande doit-il utiliser ? (0,5 pt)
6. Expliquer l'intérêt de créer une branche pour développer une nouvelle fonctionnalité sans modifier directement la branche principale. (0,5 pt)

**Exercice 3 – Interprétation d'un pipeline Jenkins (5 points)**

Une entreprise utilise Jenkins pour automatiser le processus de construction d'une application Spring Boot.

Le responsable technique propose le pipeline logique suivant :

```
Début
  |
  v
Récupération du code depuis Git
  |
  v
Compilation avec Maven
```

|
v
Exécution des tests automatiques
|
v
Préparation de la version à distribuer
|
v
Publication de l'artefact dans Nexus
|
v
Fin

Questions :

1. Quel est le rôle de Jenkins dans ce processus ? (0,5 pt)
2. Expliquer le rôle de chacune des étapes du pipeline. (2 pts)
3. Quelle est la différence entre un job Freestyle et un job Pipeline ? (1 pt)
4. Quel est l'intérêt de stocker la définition du pipeline dans un fichier Jenkinsfile versionné avec le projet ? (1 pt)
5. Que peut indiquer un échec du build Jenkins ? Donner deux causes possibles parmi les étapes présentées. (0,5 pt)

**Exercice 4 – Diagnostic d'une architecture Docker (5 points)**

Une équipe souhaite déployer une application composée d'un backend Spring Boot et d'une base de données MySQL.

Elle utilise les commandes suivantes :

```
docker pull mysql:8

docker run --name mysqldb \
--network mynetwork \
-e MYSQL_ROOT_PASSWORD=my-secret-pw \
-v /home/mysql/data:/var/lib/mysql \
-d mysql:8

docker run -p 8080:8080 \
--network mynetwork \
-d app-image-name
```

Questions :

1. Expliquer le rôle de la commande docker pull. (0,5 pt)
2. Expliquer le rôle des options suivantes :
   - --name
   - --network
   - -e
   - -v
   - -d (1,5 pt)
3. Pourquoi les deux conteneurs utilisent-ils le même réseau mynetwork ? (0,75 pt)
4. Quel est l'intérêt du volume /home/mysql/data:/var/lib/mysql ? (0,75 pt)
5. Quelle est la fonction de l'option -p 8080:8080 dans le lancement du backend ? (0,5 pt)
6. Donner les commandes permettant de :
   - Afficher les conteneurs actifs.

- Consulter les logs des services lancés avec Docker Compose.
- Arrêter les services lancés avec Docker Compose. (1 pt)

**Exercice 5 – Étude de cas : diagnostic d'une chaîne CI/CD (3 points)**

Une entreprise a mis en place une chaîne de livraison automatisée avec Git, Jenkins et Docker.
Cependant, elle rencontre les problèmes suivants :

- Jenkins ne parvient pas à récupérer le code source d'un dépôt GitHub privé.
- Le processus de compilation fonctionne, mais l'application n'est pas disponible après son déploiement.
- L'application Spring Boot démarre, mais ne parvient pas à communiquer avec la base MySQL.
- Les développeurs souhaitent publier automatiquement les images Docker construites sur Docker Hub.

Travail demandé :

Pour chaque problème :

1. Identifier une cause technique possible. (1,5 pt)
2. Proposer une solution adaptée en utilisant les outils et mécanismes étudiés dans le cours. (1,5 pt)