ESATIC
ECOLE SUPERIEURE AFRICAINE DES TIC

Novembre 2026

M1 ERIS

Durée : 02H00

Examen Usine d'automatisation -DevOps

Exercice 1 – Questions de compréhension et QCM (4 points)

Partie A – Questions de cours (2 points)

1. Définir le concept DevOps et expliquer pourquoi il ne s'agit pas uniquement d'un ensemble d'outils. (1 pt)
2. Citer deux bénéfices de l'approche DevOps pour une organisation informatique. (0,5 pt)
3. Expliquer la différence entre intégration continue (CI) et livraison/déploiement continu (CD). (0,5 pt)

Partie B – QCM (2 points)

Pour chaque question, choisir la bonne réponse.

1. Git est principalement :
   - a) Un système de gestion de base de données.
   - b) Un système de contrôle de versions distribué.
   - c) Un serveur d'intégration continue.
   - d) Un outil de conteneurisation.
2. La commande git commit permet de :
   - a) Envoyer les modifications sur GitHub.
   - b) Créer un conteneur.
   - c) Enregistrer un instantané des modifications dans le dépôt local.
   - d) Télécharger un dépôt distant.
3. Dans Jenkins, un Pipeline permet :
   - a) De configurer uniquement une base de données.
   - b) De définir les étapes d'un processus d'intégration à l'aide d'un script.
   - c) De créer des images Docker uniquement.
   - d) De remplacer Git.
4. Un conteneur Docker est :
   - a) Une instance exécutée d'une image Docker.
   - b) Un dépôt Git distant.
   - c) Un fichier de configuration Jenkins.
   - d) Une base de données relationnelle.

Exercice 2 – Maîtrise des commandes Git (4 points)

Une équipe de développement travaille sur une application Spring Boot. Le projet se trouve dans le répertoire gestion-achats.

Travail demandé :

1. Donner la commande permettant d'initialiser un dépôt Git dans ce répertoire. (0,5 pt)
2. Donner les commandes permettant de :
   - Vérifier l'état du dépôt.
   - Ajouter les fichiers src et pom.xml à l'index.

- Enregistrer les modifications avec le message « Initialisation du projet ». (1 pt)

3. Donner la commande permettant de consulter l'historique des commits. (0,5 pt)
4. Un développeur souhaite créer une branche nommée gestion-commandes. Donner les commandes permettant :
   - De créer et de basculer sur cette branche.
   - De revenir sur la branche principale master.
   - De fusionner la branche gestion-commandes dans master.
   - De supprimer la branche après fusion. (1 pt)
5. Le dépôt distant GitHub a pour adresse https://github.com/equipe/gestion-achats.git.

Donner les commandes permettant :

- D'associer le dépôt distant au dépôt local sous le nom origin.
- D'envoyer les modifications de la branche master sur GitHub. (1 pt)

Exercice 3 – Jenkins et intégration continue (5 points)

Une entreprise souhaite automatiser la compilation et les tests de son application Spring Boot à chaque modification du code source.

1. Expliquer le rôle de Jenkins dans une chaîne DevOps. (1 pt)
2. Citer les cinq grandes étapes d'un processus de build Jenkins présentées dans le cours. (1 pt)
3. Présenter deux différences entre un projet Jenkins de type Freestyle et un projet de type Pipeline. (1 pt)
4. Expliquer le rôle du fichier Jenkinsfile et préciser où il est préférable de le stocker. (1 pt)
5. Pourquoi Jenkins doit-il disposer des informations d'accès au dépôt Git lorsque celui-ci est privé ? (1 pt)

Exercice 4 – Docker et Docker Compose (4 points)

On souhaite déployer une application composée de deux services :

- Une application Spring Boot accessible sur le port 8080.
- Une base de données MySQL accessible par l'application.

Travail demandé :

1. Expliquer la différence entre une image Docker et un conteneur Docker. (1 pt)
2. Donner les commandes permettant :
   - De télécharger une image MySQL.
   - De démarrer un conteneur à partir de cette image.
   - De consulter les conteneurs actifs. (1 pt)
3. Expliquer pourquoi l'application et la base de données doivent être placées sur un même réseau Docker pour communiquer. (1 pt)
4. Donner le rôle de Docker Compose dans ce scénario. (1 pt)

Exercice 5 – Étude de cas : chaîne de déploiement (3 points)

Une équipe dispose d'une application Spring Boot dont le code est hébergé sur GitHub.

Elle souhaite mettre en place le processus suivant :

- Récupération du code source.
- Compilation avec Maven.
- Exécution des tests automatiques.
- Création d'une image Docker.
- Publication de l'image sur Docker Hub.
- Déploiement de l'application avec une base de données MySQL.

Questions :

1. Proposer un enchaînement logique des différentes étapes de cette chaîne. (1 pt)
2. Quel est le rôle de Docker Hub dans ce processus ? (0,5 pt)
3. Quel est l'intérêt d'utiliser Docker Compose pour déployer l'application et MySQL ? (0,75 pt)
4. Expliquer la différence entre la phase d'intégration continue et la phase de déploiement continu dans ce scénario. (0,75 pt)