|  |
| --- |
| **Concours DevOps**  **20 juin 2021** |

**Nom et prénom : \_\_\_\_\_\_\_\_\_\_\_AGBETIAFAN Koffi Daniel\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercice 1 5 points**

Développer les abréviations suivantes :

**Exemple :**

**SE : système d’exploitation**

* WWW :World Wide Web
* VCS :Version Control System (Systeme de Controle de Version)
* HTML :Hyper Text Mark-up Language
* IP :Internet Protocol
* DEV-OPS :Developpment OperatingSystem
* URL :Uniform Resource Locator
* CPU : Central Processing Unit
* RAM :Random Access Memory
* CI/CD :Continuous Integration(Integration Continue)/ Continuous Deployment (Deploiement Continu)
* IT :Information Technology(Technologie de l’information)

**Exercice 2 5 points**

Remplir les blancs du texte ci-dessous par les termes de la liste suivante.

**communication – déployer – vitesse – collaboration – services –**

**méthode – rapidement – développement – d’exploitation – automatisée**

DevOps est une \_\_\_\_**méthode**\_\_\_\_\_\_\_\_\_\_ de développement logiciel qui met l’accent sur la \_\_\_\_\_\_\_**communication**\_\_\_\_\_\_\_\_\_\_\_, l’intégration et la collaboration afin de permettre le déploiement rapide de produits.  
DevOps est une culture qui favorise la \_\_\_\_**collaboration**\_\_\_\_\_\_\_\_\_\_\_ entre les équipes de développement et \_\_\_\_**d’exploitation**\_\_\_\_\_\_\_\_\_\_\_. Cela permet de \_\_\_\_\_\_**déployer\_\_\_\_\_\_\_\_\_\_** le code en production plus \_\_\_\_**rapidement\_\_\_\_\_\_\_\_\_** et de manière \_\_\_\_\_\_**automatisée**\_\_\_\_\_\_\_\_\_\_\_. Cela permet d’augmenter la \_\_\_\_\_**vitesse**\_\_\_\_\_\_ d’une organisation pour fournir des applications et des \_\_\_\_\_**services**\_\_\_\_\_\_\_. Il peut être défini comme un alignement du \_\_\_\_\_\_\_**développement**\_\_\_\_\_\_\_\_\_\_\_ et des opérations informatiques.

**Exercice 3 5 points**

Questions à choix multiples. Sélectionner la ou les bonnes réponses.

1. Nous venons de créer un nouveau fichier appelé « home.html ». Lequel des éléments suivants ajoutera ce fichier afin que nous puissions le commiter dans git?

1. $ git add home.html
2. $ git add new
3. $ git add -a home.html
4. $ git git commit home.html

**A**

2. Quelle commande pour retourner à la branche principale?

1. $ git checkout origin
2. $ git checkout -b master
3. $ git checkout master
4. $ git checkout branche

3. Comment initialiser le dépôt local avec git?

1. $ git start
2. $ git pull
3. $ git clean
4. $ git init

4. Comment créer une nouvelle branche et y basculer directement?

1. $ git checkout -b <nom-de-la-branche>
2. $ git checkout < nom-de-la-branche >
3. $ git branch < nom-de-la-branche >
4. $ git branch -new < nom-de-la-branche >

###### 5. Comment vérifier l’état de votre référentiel local depuis votre dernier commit?

1. $ git check
2. $ git commit
3. $ git diff
4. $ git status

###### 6. Comment ajouter des fichiers à un commit?

1. $ git stage
2. $ git commit
3. $ git add
4. $ git reset

###### 7. Quelle commande pour vérifier l’historique de votre référentiel?

1. $ git checkout
2. $ git fetch
3. $ git log
4. $ git diff

8. Qu’est-ce qui vient en premier ?

1. $ git add
2. $ git commit

###### 9. Vous pouvez taper « git status » à tout moment dans un répertoire contrôlé par git pour vérifier le statut de vos fichiers.

1. Vrai
2. Faux

###### 10. Quelle commande vous permettra de savoir qui a modifié un fichier?

1. $ git blame
2. $ git diff
3. $ git log
4. $ git log –source

Exercice 4 10 **points (2+1+1+1+2+3)**

Répondre dans la zone encadrée

1. A quoi sert Git ?

|  |
| --- |
| **Git est un système de contrôle de version open source. Cela permet d’exécuter les versions d’un projet, qui montrent les modifications apportées au code ; permet de conserver le retour en arrière si nécessaire et d’annuler les modifications ; permet à plusieurs développeurs de vérifier et télécharger les modifications ; chaque modification peut être ensuite attribuée a un développeur spécifique.** |

1. Comment fusionner votre branche avec la branche master?

|  |
| --- |
| 1. **git checkout master** 2. **git merge <nom\_branche>** |

1. Pour supprimer une branche, quelle commande est utilisée?

|  |
| --- |
| Une fois notre branche de développement « mergée » dans la branche principale « master », nous n’avons plus besoin de la branche de développement.  Pour supprimer une branche, nous allons utiliser la commande **git branch –d <nom\_branche>** |

1. Que fait la commande « GIT PUSH » dans GIT?

|  |
| --- |
| La commande **« GIT PUSH »** met à jour les références distantes avec les objets associés. |

1. Qu’est ce que GIT GUI?

|  |
| --- |
| Le **« GIT GUI »** est l’un des meilleurs clients graphiques Git pour Linux |

1. Quelles sont les différences entre Git et Github ?

|  |
| --- |
| * **Git** est un logiciel ; il est installé localement sur notre système ; il est un outil de gestion de différentes versions des modifications apportées aux fichiers d’un référentiel git ; il fournit des fonctionnalités telle que la gestion du code source du système de contrôle de version * **Github** est un service ; il est hébergé sur le web ; il est une plateforme pour télécharger une copie de référentiel Git ; il fournit des fonctionnalités VCS() et la gestion du code source |

Exercice 4 **5 points**

Le tableau ci-dessous présente des propriétés des outils Git et Github. Associer chacune des caractéristiques suivantes à l’outil correspondant (**Git** ou bien **Github**).

| **Propriété** | **Outil** |
| --- | --- |
| C’est un logiciel | Git |
| Il est installé localement sur le système | Git |
| Il fournit une interface graphique | Git |
| C’est un outil de ligne de commande | Git |
| C’est un service | Github |
| C’est un outil de gestion de différentes versions des modifications apportées aux fichiers d’un référentiel git. | Git |
| Il fournit des fonctionnalités telles que la gestion du code source du système de contrôle de version. | Git |
| Il est hébergé sur le Web | Github |
| Il fournit des fonctionnalités de Git telles que VCS, la gestion du code source et ajoute quelques-unes de ses propres fonctionnalités. | Github |
| C’est un espace pour télécharger une copie du référentiel Git | Github |

Exercice 5  30 points (5+2+2+3+2+3+2+2+2+2+5)

1. Dans un terminal ou une fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :
2. Créer le dossier estya sur votre disque.

|  |
| --- |
| Sous word : **md estya** |

1. Se placer dans le dossier estya.

|  |
| --- |
| Sous word : **cd estya** |

1. Déterminer la version de git.

|  |
| --- |
| Sous word : **git –version** |

1. Initialiser git dans le dossier estya.

|  |
| --- |
| Sous word : **git init** en vérifiant si nous sommes dans le dossier estya |

1. Vérifier dans la configuration le nom de l’utilisateur et son adresse mail.

|  |
| --- |
| 1. Vérification de l’identité nom dans la configuration : **git config --global user.name ‘’nom\_utilisateur’’** 2. Vérification de l’identité email dans la configuration : **git config --global user.email ‘’email\_utilisateur’’** |

1. Créer le fichier index.html dans le dossier estya. Y mettre le code ci-dessous.
2. <!DOCTYPE html>
3. <html lang="en">
4. <head>
5. <meta charset="UTF-8" />
6. <meta http-equiv="X-UA-Compatible" content="IE=edge" />
7. <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8. <title>Formation DevOps</title>
9. </head>
10. <body>
11. <h1>Plan</h1>
12. <p>Introduction à DevOps</p>
13. <p>Outil VCS : git et github</p>
14. <p>Collaboration avec github</p>
15. <p>CI/CD avec github actions</p>
16. </body>
17. </html>
18. Créer le fichier intro.html dans le dossier estya. Y mettre le code ci-dessous.
19. <!DOCTYPE html>
20. <html lang="en">
21. <head>
22. <meta charset="UTF-8" />
23. <meta http-equiv="X-UA-Compatible" content="IE=edge" />
24. <meta name="viewport" content="width=device-width, initial-scale=1.0" />
25. <title>DevOps</title>
26. </head>
27. <body>
28. <h1>Introduction</h1>
29. <h2>Définition</h2>
30. <p>
31. DevOps est un mouvement en ingénierie informatique et une pratique
32. technique visant à l'unification du développement logiciel (dev) et de
33. l'administration des infrastructures informatiques (ops), notamment
34. l'administration système.
35. </p>
36. <p>
37. Certaines définitions « commerciales » tentent de coupler DevOps à des
38. outils de construction de logiciel ou à des infrastructures Cloud.
39. </p>
40. <p>
41. D’autres définitions trompeuses voient DevOps comme une intersection du
42. travail des hommes dans une organisation IT.
43. </p>
44. </body>
45. </html>
46. Dans le terminal ou la fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :
47. Sauvegarder l’état actuel du dossier. Utiliser un commentaire de votre choix.

|  |
| --- |
| Elle se fait en deux(02) étapes :   * Sélection de mes deux fichiers (index et intro) : **git add .** * Enregistrement de mes deux fichiers : **git commit –m ‘’Initialisation du projet’’** |

1. Créer une nouvelle branche hyperLinks.

|  |
| --- |
| * Création d’une nouvelle branche : **git branch hyperLinks** |

1. Basculer sur la branche nouvellement créée.

|  |
| --- |
| * Bascule sur la nouvelle branche : **git checkout hyperLinks** |

1. Modifier le fichier index.html. remplacer la ligne 13 par le code ci-dessous.

<p><a href="./intro.html">Introduction à DevOps</a></p>

1. Dans le terminal ou la fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :
2. Faire un commit des modifications apportées.

|  |
| --- |
| Toujours en deux étapes :   * Reselectionner le fichier subit par la modification : **git add .** * Enregistrer à nouveau les modifications : **git commit –m ‘’Modification de la ligne 13 dans le fichier index’’** |

1. Rapatrier les modifications sur la branche master.

|  |
| --- |
| En deux etapes :   * Retour sur la branche master d’abord : **git checkout master** * Fusion des modifications : **git merge hyperLinks** |

1. Supprimer la branche hyperLinks.

|  |
| --- |
| * Suppression de la branche hyperLinks : **git branch –d hyperLinks** |

1. Copier ce document, après avoir répondu aux questions, dans votre dépôt local (votre dossier estya)
2. Dans votre compte Github créer un repository Estya
3. Exporter votre dépôt local sur le dépôt distant.
4. M’inviter pour pouvoir accéder à votre dépôt distant maher.jaouadi70@gmail.com
5. Donner une définition de github actions. En quoi est utile ?

|  |
| --- |
| Le github actions est une suite d’actions notamment une première qui permet de construire notre projet et de dérouler un ensemble de tests unitaires, d’intégration et de sécurité que l’on utilisera ensuite dans un flux de travail(workflow) ; une deuxième action qui peut être rajouter au flux de travail permettant de publier les résultats sur la file de discussion en y appliquant des labels comme Bug, non-secure, code-sale,… une troisième action réalisant des demandes et en les réutilisant au niveau des flux de travail.  L’utilité de github actions :  . il est un serveur embarqué (local)  . il nous libère des taches de gestion et de maintenance de l’environnement d’exécution et du serveur CI/CD  . de loin la plus facile a utiliser et celle qui requiert le moins de compétences techniques. |

**Bon travail**