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

**Nom et prénom : \_NDIAYE Samba\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercice 1 5 points**

Développer les abréviations suivantes :

**Exemple :**

**SE : système d’exploitation**

* WWW : World Wide Web
* VCS : Version Control System
* HTML : HyperText Markup Langage
* IP : Internet Protocol
* DEV-OPS : Development Operations
* URL : Uniform Resource Locator
* CPU : Central Processing Unit
* RAM : Random Access Memory
* CI/CD : Continuous Integration/Continuous Deployment
* IT : Information Technology

**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 outil qui permet qui permet de gérer différents projets en les envoyant sur un serveur en offrant la possibilité de voir leurs historiques ainsi que la traçabilité de toutes les modifications qui y sont apportées. |

1. Comment fusionner votre branche avec la branche master?

|  |
| --- |
| Revenir à la branche master avec git checkout master puis git merge « nom branche » pour la fusion |

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

|  |
| --- |
| git branch -d « nom branche » |

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

|  |
| --- |
| La commande « GIT PUSH » permet de **déposer le projet sur le dépôt distant**. |

1. Qu’est ce que GIT GUI?

|  |
| --- |
| GIT GUI est une interface graphique de GIT permettant de créer un nouveau dépôt Git en local en indiquant le nom de celui-ci et de cloner un dépôt distant en local en indiquant le chemin vers la source et le dossier de destination. |

1. Quelles sont les différences entre Git et Github ?

|  |
| --- |
| **Git** est un outil de gestion de version alors que **Github** est une plateforme en ligne qui permet d’héberger les dépôts Git. |

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. | **Git** |
| 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.

|  |
| --- |
| mkdir estya |

1. Se placer dans le dossier estya.

|  |
| --- |
| cd estya |

1. Déterminer la version de git.

|  |
| --- |
| git --version |

1. Initialiser git dans le dossier estya.

|  |
| --- |
| git init |

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

|  |
| --- |
| git log |

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.

|  |
| --- |
| git commit -m "estya initial" |

1. Créer une nouvelle branche hyperLinks.

|  |
| --- |
| git branch hyperLinks |

1. Basculer sur la branche nouvellement créée.

|  |
| --- |
| 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.

|  |
| --- |
| git add .  puis  git commit -m "estya deux" |

1. Rapatrier les modifications sur la branche master.

|  |
| --- |
| git checkout master puis git merge hyperLinks |

1. Supprimer 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 ?

|  |
| --- |
| **Github Actions** est le nouveau serveur build de Github qui permet d’automatiser votre flux de travail directement à partir de votre référentiel Github, sans avoir à configurer et à gérer un serveur tiers comme jenkins, Bamboo, CircleCI.  Utilité de **Github Actions** :   * Automatisation du flux de travail des équipes de développement * Possibilité pour les utilisateurs de construire, tester et déployer leur code directement à partir de GitHub * Créer des fonctionnalités d’intégration continue (CI) et de déploiement continu (CD) de bout en bout directement à partir d’un référentiel GitHub |

**Bon travail**