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

**Nom et prénom : AGORO Jocelyn**

**Exercice 1 5 points**

Développer les abréviations suivantes :

**Exemple :**

**SE : système d’exploitation**

* WWW : World Wide Web
* VCS : Version Control Systems
* HTML : HyperText Markup Language
* IP : Internet Protocol
* DEV-OPS : Development-Operations
* URL : Uniform Resource Locator
* CPU : Central Processing Unit
* RAM : Random-Access Memory
* CI/CD : Continuous Integration/Continuous Delivery
* 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 de versionning, c’est-à-dire un outil qui nous permet de gérer les différentes versions de nos codes en y créant des sauvegardes. Il permet alors de garder un historique clair des différentes versions de nos produits que nous avons commiter. |

1. Comment fusionner votre branche avec la branche master?

|  |
| --- |
| Si la branche existe déja et s’appelle branch\_name :  $ git checkout master  $ git merge branch\_name  Sinon :  $ git branch branch\_name  $ git checkout master  $ git merge branch\_name |

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

|  |
| --- |
| Supposons que la branche s’appelle branch\_name :  $ git branch -d branch\_name  ou  $ git branch -D branch\_name |

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

|  |
| --- |
| $ git push origin <branch>  git push permet de téléverser notre dépôt local sur un dépôt distant github afin de permettre la collaboration avec d’autres intervenants sur un projet. |

1. Qu’est ce que GIT GUI?

|  |
| --- |
| GIT GUI est la version avec interface graphique de GIT qui nous permet d’effectuer certaines tâches, non pas avec des commandes, mais avec des boutons qui exécutent derrière les commandes GIT CLI. |

1. Quelles sont les différences entre Git et Github ?

|  |
| --- |
| Git est un version control system qui permet de gérer et de garder les traces de l’historique de nos codes sources.Github est un service orienté cloud qui facilite la gestion des dépôts Git.  Git est un version control system local et Github est une plateforme web qui intègre des fonctionnalités du version control de git.  Git donne accès a une CLI et Github à une Interface graphique intuitive. |

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. | **Github** |
| Il fournit des fonctionnalités telles que la gestion du code source du système de contrôle de version. | **Github** |
| 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.

|  |
| --- |
| $ 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 config --global -l |

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.

|  |
| --- |
| $ cd estya  $ git init  $ git add .  $ git status  $ git commit -m " première version " |

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 index.html  $ git commit -m " ajout d’un lien à ‘ Introduction à DevOps ’ " |

1. Rapatrier les modifications sur la branche master.

|  |
| --- |
| $ git checkout master  $ 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 de build de GitHub, à l’image du pipeline gitlab-ci de Gitlab.  Il permet de gérer les workflow à partir de script s’effectuant par le déclenchement de certains évènements de git comme push,pull, ... . Il permet d’automatiser votre flux de travail directement à partir de votre référentiel git sans avoir à gérer des serveurs tiers tel que Jenkins, Bamboo, CircleCI et autres. Il présente aussi l’avantage d’être embarqué près de Github ce qui facilite les tâches et la gestion des serveurs tiers si on en utilisait des serveurs comme Jenkins par exemple. Sa caractéristique la plus intéressante est sans doute l’automatisation des tâches grâce à des scripts assez simple à comprendre et une communauté assez importante qui permet de maîtriser rapidement la syntaxe ou qui met à disposition des exemples de script gratuitement pour aider accélérer les production. Il dispose aussi d’alternative pour automatiser nos tâches avec des outils si nous en maîtrisons déjà comme ansible,puppet et bien d’autres. |

**Bon travail**