

**20 juin 2021**

**Concours DevOps**

# Nom et prénom : Ahmed Aziz Attia \_

**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 : développement logiciel (dev) et administration des infrastructures informatiques (ops)
* URL : Uniform Resource Locator
* CPU : central processing unit
* RAM : Random-access memory
* CI/CD : the combined practices of continuous integration and either continuous delivery or 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

1. 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
2. Comment initialiser le dépôt local avec git?
   1. $ git start
   2. $ git pull
   3. $ git clean
   4. $ git init
3. 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 >
4. Comment vérifier l’état de votre référentiel local depuis votre dernier commit?
   1. $ git check
   2. $ git commit
   3. $ git diff
   4. $ git status
5. Comment ajouter des fichiers à un commit?
   1. $ git stage
   2. $ git commit
   3. $ git add
   4. $ git reset
6. Quelle commande pour vérifier l’historique de votre référentiel?
   1. $ git checkout
   2. $ git fetch
   3. $ git log
   4. $ git diff
7. Qu’est-ce qui vient en premier ?
   1. $ git add
   2. $ git commit
8. 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
9. 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 ?

C’est un logiciel de versioning qui permet de conserver un historique des modifications effectuées sur un projet afin de pouvoir rapidement identifier les changements effectuées et de revenir à une ancienne version en cas de problème

* + 1. Comment fusionner votre branche avec la branche master?

Nous exécutons d'abord git checkout master pour remettre la branche active en master. Ensuite, nous exécutons la commande git merge new-branch pour fusionner la nouvelle fonctionnalité dans la branche master.

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

git branch -d <branche>

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

Met à jour les références distantes à l’aide de références locales, tout en envoyant les objets nécessaires pour terminer les références données.

* + 1. Qu’est ce que GIT GUI?

Git GUI est un **G**raphical **U**ser **I**nterface vous permettant d’utiliser Git sans toucher à la ligne de commande. C’est une alternative parmi les autres clients Git

* + 1. Quelles sont les différences entre Git et Github ?

Git est un système de contrôle de version qui vous permet de gérer et de suivre l’historique de votre code source. GitHub est un service d’hébergement basé sur le cloud qui vous permet de gérer les référentiels Git GitHub est un cloud-based hosting service qui vous permet de gérer Git référentiels. Si vous avez des projets open source qui utilisent Git, GitHub est conçu pour vous aider à mieux les gérer.

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 | Github |
| 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 :
   1. 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 --list --show-origin

$ git config --global user.name "Ahmed Aziz Attia"

$ git config --global user.email ”AzizCMakarov@gmail.com”

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" /> 9- <title>Formation DevOps</title>
8. </head>
9. <body>

12- <h1>Plan</h1>

1. <p>Introduction à DevOps</p>
2. <p>Outil VCS : git et github</p> 15- <p>Collaboration avec github</p> 16- <p>CI/CD avec github actions</p> 17- </body>

18- </html>

19-

1. Créer le fichier intro.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" /> 10- <title>DevOps</title>
8. </head>
9. <body>
10. <h1>Introduction</h1>
11. <h2>Définition</h2>

15- <p>

1. DevOps est un mouvement en ingénierie informatique et une pratique
2. technique visant à l'unification du développement logiciel (dev) et de 18- l'administration des infrastructures informatiques (ops), notamment

19- l'administration système.

20- </p>

21- <p>

22- Certaines définitions « commerciales » tentent de coupler DevOps à des 23- outils de construction de logiciel ou à des infrastructures Cloud.

24- </p>

25- <p>

26- D’autres définitions trompeuses voient DevOps comme une intersection du 27- travail des hommes dans une organisation IT.

28- </p>

1. </body>
2. </html>
3. Dans le terminal ou la fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :
4. Sauvegarder l’état actuel du dossier. Utiliser un commentaire de votre choix.

git add .

git commit -m  ‘’Question f’’

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  ‘’Modification de la ligne 13’’

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 9- Exporter votre dépôt local sur le dépôt distant.

10- M’inviter pour pouvoir accéder à votre dépôt distant [maher.jaouadi70@gmail.com](mailto:maher.jaouadi70@gmail.com) 11- 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 créer des workflows de cycle de vie de développement logiciel personnalisés directement dans votre référentiel Github. Ces flux de travail sont constitués de différentes tâches appelées actions qui peuvent être exécutées automatiquement sur certains événements. Cela vous permet d’inclure des fonctionnalités d’intégration continue (CI) et de déploiement continu (CD) ainsi que de nombreuses autres fonctionnalités directement dans votre référentiel. Il apporte une culture d’automatisation en définissant un ensemble de bonnes pratiques permettant d’automatiser certaines tâches redondantes.

On offre ainsi la possibilité de créer des fonctionnalités d’intégration continue (CI) et de déploiement continu (CD) de bout en bout directement à partir d’un référentiel GitHub, sans faire recours à un serveur tiers comme Jenkins, Bamboo, CircleCI ou autres.

# Bon travail