



Concours DevOps

20 juin 2021

Nom et prénom : RACHEDI Mokrane

Exercice 1

5 points

Développer les abréviations suivantes :

Exemple :

SE : système d'exploitation

- WWW : World Wide Web
- VCS : Version Control System
- HTML : HyperText Markup Language
- IP : Internet Protocol
- DEV-OPS : DEV -> Development, OPS -> 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?

- a. **\$ git add home.html**
- b. \$ git add new
- c. \$ git add -a home.html
- d. \$ git git commit home.html

A

2. Quelle commande pour retourner à la branche principale?

- a. \$ git checkout origin
- b. \$ git checkout -b master
- c. **\$ git checkout master**
- d. \$ git checkout branche

3. Comment initialiser le dépôt local avec git?

- a. \$ git start
- b. \$ git pull
- c. \$ git clean
- d. **\$ git init**

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

- a. **\$ git checkout -b <nom-de-la-branche>**
- b. \$ git checkout < nom-de-la-branche >
- c. \$ git branch < nom-de-la-branche >
- d. \$ git branch -new < nom-de-la-branche >

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

- a. \$ git check
- b. \$ git commit
- c. \$ git diff
- d. **\$ git status**

6. Comment ajouter des fichiers à un commit?

- a. \$ git stage
- b. \$ git commit
- c. **\$ git add**
- d. \$ git reset

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

- a. \$ git checkout
- b. \$ git fetch
- c. **\$ git log**
- d. \$ git diff

8. Qu'est-ce qui vient en premier ?

- a. `$ git add`
- b. `$ git commit`

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

- a. Vrai
- b. Faux

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

- a. `$ git blame`
- b. `$ git diff`
- c. `$ git log`
- d. `$ 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 qui permet de suivre, dans le temps, les modifications apportées à notre projet (avoir un historique de toutes les versions de notre projet).

2. Comment fusionner votre branche avec la branche master?

```
$ git checkout master  
$ git merge <nom_de_notre_branche>
```

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

```
$ git branch -d <nom_de_la_branche>
```

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

GIT PUSH met à jour les références distantes avec les objets associés. c'est-à-dire on va déposer (pousser) les modifications qu'on a apportées à notre projet (en local) vers le dépôt distant (GitHub).

5. Qu'est ce que GIT GUI?

C'est une interface graphique du logiciel Git.

6. Quelles sont les différences entre Git et Github ?

Git est le système de contrôle de versions, tandis que GitHub est un service qui héberge des dépôts Git et aide les utilisateurs à collaborer pour le développement de logiciels (GitHub est un service hébergé sur le Web).

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+5)

1- Dans un terminal ou une fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :

a. Créer le dossier estya sur votre disque.

```
$ mkdir estya
```

b. Se placer dans le dossier estya.

```
$ cd estya
```

c. Déterminer la version de git.

```
$ git --version
```

- d. Initialiser git dans le dossier estya.

```
$ mkdir estya
$ cd estya
$ git init
```

- e. Vérifier dans la configuration le nom de l'utilisateur et son adresse mail.

```
$ git config --global -l
```

- 2- Créer le fichier index.html dans le dossier estya. Y mettre le code ci-dessous.

```
3- <!DOCTYPE html>
4- <html lang="en">
5-   <head>
6-     <meta charset="UTF-8" />
7-     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
8-     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
9-     <title>Formation DevOps</title>
10-  </head>
11-  <body>
12-    <h1>Plan</h1>
13-    <p>Introduction à DevOps</p>
14-    <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-
```

- 3- Créer le fichier intro.html dans le dossier estya. Y mettre le code ci-dessous.

```
4- <!DOCTYPE html>
5- <html lang="en">
6-   <head>
7-     <meta charset="UTF-8" />
8-     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
9-     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
10-    <title>DevOps</title>
11-  </head>
12-  <body>
13-    <h1>Introduction</h1>
14-    <h2>Définition</h2>
15-    <p>
16-      DevOps est un mouvement en ingénierie informatique et une pratique
17-      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>
29-   </body>
30- </html>
31-
```

- 4- Dans le terminal ou la fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :

- f. Sauvegarder l'état actuel du dossier. Utiliser un commentaire de votre choix.

```
$ git add .
$ git commit -m "Initialisation du projet"
```

- g. Créer une nouvelle branche hyperLinks.

```
$ git branch hyperLinks
```

- h. Basculer sur la branche nouvellement créée.

```
$ git checkout hyperLinks
```

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

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

- 6- Dans le terminal ou la fenêtre invité de commande donner la ou les commandes correspondantes aux actions suivantes :

- i. Faire un commit des modifications apportées.

```
$ git add .
$ git commit -m "Modification de la ligne 11 de index.html"
```

- j. Rapatrier les modifications sur la branche master.

```
$ git checkout master
$ git merge hyperLinks
```

- k. Supprimer la branche hyperLinks.

```
$ git branch -d hyperLinks
```

- 7- Copier ce document, après avoir répondu aux questions, dans votre dépôt local (votre dossier estya)
- 8- 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
- 11- Donner une définition de github actions. En quoi est utile ?

GitHub Actions est le nouveau serveur de construction de build de GitHub, à l'image du pipeline gitlab-ci de Gitlab. Il permet d'automatiser notre flux de travail directement à partir de notre référentiel GitHub, sans avoir à configurer et à gérer un serveur tiers comme jenkins, Bamboo, CircleCI.

Il apporte une culture d'automatisation en définissant un ensemble de bonnes pratiques permettant d'automatiser certaines tâches redondantes qu'on faisait autrefois manuellement.

Il offre 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