



Concours DevOps

20 juin 2021

Nom et prénom : _____ Pascal Babou Sarr _____ (sarmelo@gmail.com)

Exercice 1

5 points

Développer les abréviations suivantes :

Exemple :

SE : système d'exploitation

- WWW : World Wide Web
- VCS : Version Control Software.
- HTML : Hypertext Markup Language
- IP : Internet Protocol
- DEV-OPS : Development and Operations
- URL : Uniform Resource Locator
- CPU : Central Processing Unit
- RAM : Random Access Memory
- CI/CD : Continuous Integration / Continuous delivery
- IT : Information Technologie

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 ?

Git un Est outil ou logiciel qui permet de gérer ou d'envoyer des projets sur un serveur distant comme github.

2. Comment fusionner votre branche avec la branche master?

Supposons que la branche backgroundColor est crée et qu'on veut fusionner :

- Git checkout master
- Git merge backgroundColor (fusion de la branche)

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

Git branch -d <nomdelabranche>

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

Il est utilisée pour pousser un commit depuis la branche locale vers la branche distante

5. Qu'est ce que GIT GUI?

Interface Graphique , outil permettant les opérations Git courantes.

6. Quelles sont les différences entre Git et Github ?

Git	Github
Logiciel local	Plateforme web (service)
Outil de ligne de commande	Interface Graphique
Gestion des differentes versions des modifications	espace pour telecharger copie referentiel
Fournit des fonctionnalités gestion code source	fournit des fonctionnalités de git tel que VCS

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 :

a. Créer le dossier estya sur votre disque.

Sur windows :

`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.

On se positionne sur estya puis : **git init**

Message de retour : `Initialized empty Git repository in C:/Users/sarm/pascal/estya/.git/`

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

git config --global -l

Message de retour :

`user.email=sarmelo@gmail.com
user.name=Pascal sarr`

- 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.

Pour sauvegarder on fait :

- **git add** . pour prendre en compte les deux fichiers
 - **git status** .pour vérifier si les fichiers sont placés dans le staging area.
- Pour terminer - **git commit -m "ajout de index.html et intro.html"**

Résultat : **[master (root-commit) b372490] ajout de index.html et intro.html**
2 files changed, 43 insertions(+)
create mode 100644 index.html
create mode 100644 intro.html

g. Créer une nouvelle branche hyperLinks.

Création de la branche - **git branch hyperLinks**

Git branch pour vérifier la création :

C:\Users\sarm\pascal\estya> **git branch**

```

hyperLinks.
*master.

```

h. Basculer sur la branche nouvellement créée.

Bascule - **git checkout hyperLinks**
Switched to branch 'hyperLinks'

Vérification

C:\Users\sarm\pascal\estya> **git branch**

```

*hyperLinks
master

```

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.

Après avoir modifié on fait un git diff pour vérifier la modification :

- `git diff` (Optionel)

Resultat : `<p>Introduction <C3><A0> DevOps</p> .`

- `git add .`
- `git status` (Optionel) pour vérifier si c'est modifié.
- `git commit -m "Modification ligne 13 index.html"`
- `git log` (Optionel)

Resultat:

```
C:\Users\sarm\pascal\estya> git log
commit 7ac2c644d91f2f597c819257edccec64238f8a55 (HEAD -> hyperLinks)
Author: Pascal sarr <sarmelo@gmail.com>
Date: Sun Jun 20 12:06:59 2021 +0000
Modification ligne 13 index.html
```

j. Rapatrier les modifications sur la branche master.

- `git checkout master`
- `git merge hyperLinks`

Resultat :

```
C:\Users\sarm\pascal\estya> git merge hyperLinks
Updating b372490..7ac2c64
Fast-forward
index.html | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

k. Supprimer la branche hyperLinks.

Avant :

```
C:\Users\sarm\pascal\estya> git branch
hyperLinks
* master
```

```
C:\Users\sarm\pascal\estya> git branch -d hyperLinks
Deleted branch hyperLinks (was 7ac2c64).
```

Vérification soit par : `git branch`

```
* master
```

- 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 une solution d'intégration continue et de déploiement automatisée(CI/CD), une extension de GitHub.

Il nous permet d'écrire un processus à lancer en rapport avec son projet et qui sera exécuter en fonction des critères définis dans l'action à réaliser .
nous permet d'automatiser facilement nos workflows logiciels grâce a notre CI/CD

Bon travail