



# Concours DevOps

20 juin 2021

Nom et prénom : \_GONNON Deiss\_Kurt de s-p.\_

## 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 : Development Operations
- URL : Uniform Resource Locator
- CPU : Central Processing Unit
- RAM : Random Access Memory
- CI/CD : Continuous Integration / Continuous Development
- 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 **automatisé**. 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 permet de garder toutes les versions de son projet, il permet de connaître l'historique de ce qui a été fait, qui l'a fait, quand et pourquoi cela a été fait. C'est en général un système de gestion de version décentralisé.

2. Comment fusionner votre branche avec la branche master?

```
$ git checkout master #Si l'on est pas sur la branche principale  
$ git merge <nom-de-la-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?

Elle permet d'envoyer les fichiers du projet de son dépôt local vers son dépôt distant. Ainsi on met à jour le référentiel distant ainsi que les objets associés

5. Qu'est-ce que GIT GUI?

GIT GUI est le logiciel à interface graphique du logiciel git permettant d'effectuer les mêmes opérations possible en ligne de commande avec git.

6. Quelles sont les différences entre Git et Github ?

Git est logiciel de gestion de version, GitHub est la plateforme web de ce logiciel offrant des services d'hébergement de code source, de contrôle de version distribué, et de gestion de code source de git.

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	<b>Git</b>
Il est installé localement sur le système	<b>Git</b>
Il fournit une interface graphique	<b>GitHub</b>
C'est un outil de ligne de commande	<b>Git</b>
C'est un service	<b>GitHub</b>
C'est un outil de gestion de différentes versions des modifications apportées aux fichiers d'un référentiel git.	<b>Git</b>
Il fournit des fonctionnalités telles que la gestion du code source du système de contrôle de version.	<b>Git</b>
Il est hébergé sur le Web	<b>GitHub</b>
Il fournit des fonctionnalités de Git telles que VCS, la gestion du code source et ajoute quelques-unes de ses propres fonctionnalités.	<b>GitHub</b>
C'est un espace pour télécharger une copie du référentiel Git	<b>GitHub</b>

1- Dans un terminal ou une fenêtre invitée 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.

```
$git init
```

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

```
$git config --global user.name  
$git config --global user.email
```

- 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 "Mofification de index.html et intro.html"
```

- 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 du fichier index.html. Remplacement de la ligne 13 "
```

- 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](mailto:maher.jaouadi70@gmail.com)  
11- Donner une définition de github actions. En quoi est utile ?

GitHub Action est le serveur de construction embarqué à gitHub pour les CI/CD. C'est une partie d'un projet gitHub permettant d'automatiser certaines tâches redondantes qui se faisait manuellement (cause de perte de temps de retard dans les projet).

Ainsi avec gitHub action par exemple ont peut mettre en place un processus de validation de code lorsqu'un push est effectué.

GitHub Actions permet de lancer des machines Linux / Windows / MacOS pour faire de l'intégration en continu (déployer son code) de manière automatique et sans coût supplémentaire de licence.

**Bon travail**