Linux

1 - Cherchez s’il existe un logiciel installable appelé mysql. Il faut que le nom contienne « mysql », pas que mysql apparaisse dans la description.

apt search mysql --names-only

2 - Les deux paquets qui nous intéressent en fait sont mysql-client et mysql-server. Vérifiez qu’ils sont bien tous les deux disponibles.

apt search mysql-client --names-only

apt search mysql-serveur --names-only

3 - Vérifiez que la version majeure qui sera installée avec ces deux paquets est bien la dernière disponible.

apt info mysql-client

apt info mysql-serveur

4 - Installez les deux paquets (vous en aurez besoin pour le cours de BD)

sudo apt install mysql-client

sudo apt install mysql-server

5 - Faites la même chose avec le paquet postgresql (dont vous aurez aussi besoin pour le cours de BD).

sudo apt install postgresql

6 - En fait, la version installée n’est pas la dernière majeure (la 14). Du coup désinstallez-là,

**SI**

7 - Cherchez sur internet la procédure officielle pour récupérer le dépot posgresql pour Ubuntu. Vous devriez tomber sur : https://www.postgresql.org/download/linux/ubuntu/. (Faites l’effort de chercher sur internet, ca fait parti des compétences à avoir). Ne suivez pas bêtement la procédure, on va la refaire à la main.

**NON**

8 - La première chose à faire est de modifier le fichier sources.list. Pour cela, tapez la commande « sudo nano /etc/apt/sources.list ». L’éditeur de texte en ligne « nano » devrait s’ouvrir avec le contenu du fichier.

sudo nano /etc/apt/sources.list

9 - Dans la page web décrivant la procédure d’installation du dépot, on recherche une ligne de la forme « deb http://.. main». C’est la ligne qu’il faudra recopier à la fin du fichier sources.list. Cette ligne comprend une variable « $(lsb\_release -cs) » à remplacer par le nom de la version d’Ubuntu (focal). Ajoutez donc à la fin du fichier sources.list la ligne : deb http://apt.postgresql.org/pub/repos/apt focal-pgdg main

10 - Pour sauvegarder le fichier, appuyez sur la touche « controle » puis sur la touche « O » (ce qui s’écrit, en raccourci : ctrl-O ou encore ^-O). Acceptez le nom de fichier par défaut et appuyez sur « entrée ».

11 - En fait la bande du bas du terminal est un pense-bete qui rappelle les principales commandes. Trouvez la commande pour quitter nano.

CTRL + S

CTRL + X

12 - Pour éviter que n’importe qui se fasse passer pour le dépot officiel d’un logiciel, les dépots sont signés. Pour importer la signature du nouveau dépot, tapez la commande indiquée dans la documentation en ligne : « wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add - » Vous devriez obtenir un « OK » sur la ligne de commande en réponse.

13 - Mettez à jour en demandant à apt de relire ses dépots, vérifiez que c’est maintenant la dernière version de postgresql qui sera installée (paquet postgresql-14) et installez-la.

sudo apt update

sudo apt upgrade

15 - Faites la même chose avec MongoDB

**NON**

16 - Profitez-en pour mettre votre système à jour. Si vous faites cela régulièrement, cela prendra moins de temps. Pensez-y.

**NON**

17 - Faites la même chose avec pip3 pour le paquet pygame (attention, pip3 search ne fonctionne plus).

**NON**

GIT

1 - Récupérez le répertoire git du projet vide que vous venez de créer.

2 - Dans un autre répertoire, récupérez le code source du projet Flash : https://github.com/pallets/flask

3 - Recopiez dans le projet vide le répertoire « src » du projet Flask.

4 - Vérifiez l’état de votre projet git.

5 - Créez ainsi une première version de votre projet (sans oublier d’ajouter d’abord l’intégralité du répertoire « src » à l’index).

6 - Vérifiez l’état de votre projet git.

7 - Vérifiez que le commit a bien été créé.

8 - Allez dans le répertoire « src/flask » et modifiez le fichier « blueprints.py » : modifiez le champs « app » de « Flask » à « Flaque ».

9 - Vérifiez l’état des fichiers de votre projet.

10 - Vérifiez les modifications que vous avez faites par rapport à la dernière version.

11 - Modifiez le fichier « helpers.py » en remplacant « FLASK\_ENV » par « FLAQUE\_ENV ».

12 - Vérifiez l’état des fichiers de votre projet.

13 - Vérifiez les modifications que vous avez faites par rapport à la dernière version.

14 - Créez une nouvelle version.

15 - Vérifiez l’état de votre projet git.

16 - Vérifiez que le commit a bien été créé.

17 - Poussez les nouvelles modifications sur le serveur gitlab.

18 - Refaites la manip 2 fois, en changeant si vous voulez les fichiers et les modifications faites. N’oubliez pas de bien vérifier l’état des fichiers, l’état du log, les différences depuis la dernière version. Le but est que cet suite d’étape soit totalement naturel pour vous et que vous puissiez enchainer ces commandes sans avoir à réfléchir