13/12/2020

Valentin Gratz

Annuaire – Book lignes de commandes VPS/GIT T

Sommaire

[Lignes de commandes GIT 2](#_Toc60501330)

[En local 2](#_Toc60501331)

[Configuration du profil 2](#_Toc60501332)

[Le terminal 2](#_Toc60501333)

[Les branches 2](#_Toc60501334)

[Le statut 2](#_Toc60501335)

[Snapshot 2](#_Toc60501336)

[ChangeLog 2](#_Toc60501337)

[Fusion de branches 2](#_Toc60501338)

[Les conflits 3](#_Toc60501339)

[Faire le ménage 3](#_Toc60501340)

[En remote (github, gitlab…) 3](#_Toc60501341)

[Lignes de commandes VPS Linux 4](#_Toc60501342)

[Changer mot de passe sur un VPS 4](#_Toc60501343)

[Les différents dossiers les plus importants 4](#_Toc60501344)

[Ajouter un utilisateur 5](#_Toc60501345)

[Gérer les permissions des fichiers / répertoires 5](#_Toc60501346)

[Vérifier les mises à jour des paquets 6](#_Toc60501347)

[Raccourcis clavier pour coder plus vite 7](#_Toc60501348)

[Lignes pour Compozer 8](#_Toc60501349)

[Lignes de commandes Symfony 9](#_Toc60501350)

[User et authentification 9](#_Toc60501351)

[Configuration de twig avec bootstrap 9](#_Toc60501352)

# Lignes de commandes GIT

Toujours mettre $git avant la commande à saisir.

Pour coller, toujours faire un clic droit, et jamais le raccourci clavier, comme pour les VPS.

## En local

### Configuration du profil

Installation de git sur l’ordinateur : [www.git-scm.com](http://www.git-scm.com) (pour windows)

$git config --global user.name « Prénom Nom »

$git config --global user.email «mon email »

### Le terminal

Win + R, saisir « cmd »

Ou faire un clic droit dans le dossier et cliquer sur « Git Bash Here ».

### Les branches

$git branch // liste toutes les branches du projet, dont la « master » par défaut.

$git branch « nom » // pour créer une branche

$git checkout « nom » // pour se déplacer d’une branche à une autre

### Le statut

$git status // pour obtenir le status du git

### Snapshot

Commit instantanée du projet à un temps donné.

$git add (nom du fichier) ou « . »

$git commit -a-m « Mon message de commit »

### ChangeLog

$git diff // liste les modifications effectuées, affiche le code ajouté

$git log // liste les id/auteur des modifications

### Fusion de branches

En premier, il faut s’assurer d’être sur la branche MASTER

$git diff master*(branche1)*..branche 2

**q // pour sortir de l’historique des modifications**

**si ça ne marche pas, essayer Ctrl C ou quitter la console, sinon**

**fermer git et rouvrir**

$git merge nombranche pour voir les updates.

### Les conflits

Si un conflit apparaît, régler manuellement et choisir entre les codes affichés.

### Faire le ménage

Supprimer une branche quand elle n’est plus utile.

$git branche **-d** nomdelabranche // d’abord s’assurer qu’elle à été merge.

$git branche **-D** nomdelabranche // supprime même si elle n’à pas été merge.

$clear // pour nettoyer la console.

## En remote (github, gitlab…)

Créer d’abord le reposity/projet sur github ou un autre site. Pour envoyer le dossier local vers le site (github), faire cette commande :

$git remote add origin url.git

(ajouter) (origin = branche master) et le lien du repo git.

$git push « ou j’envoie » « sur quelle branche »

Origin master (par exemple)

$git pull « où je récupère » « sur quelle branche » // transfère le local sur github

Pour cloner un reposity github, copier le lien HTTPS.

Pour récupérer le contenu dans un dossier, toujours faire un **$git init**. Ensuite **$git remote add origin**.

# Lignes de commandes VPS Linux

### Changer mot de passe sur un VPS

Passwd root // ligne de commande

|  |  |
| --- | --- |
| Nom de la commande | A quoi elle correspond |
| ls | Affichage du contenu d’un répertoire |
| touch | Utilisé en général pour créer un fichier |
| cat | Affiche-le contenue d’un fichier (txt) |
| cd | Changement de répertoire courant |
| cp | Copie |
| mv | Déplacement, renommage |
| pwd | Affichage répertoire courant |
| man | Aide en ligne pour les commandes |
| rm | Suppression fichier ou répertoire vide (+ -R pour répertoire ; -f sans demander d’autorisation) |
| mkdir | Création répertoire |
| rm – R | Suppression répertoire |
| adduser | Ajouter un utilisateur |
| addgroup | Ajouter un groupe d’utilisateur |
| passwd <user> | Modifier le mot de passe d’un utilisateur |

### Les différents dossiers les plus importants

|  |  |
| --- | --- |
| Nom du dossier | A quoi il correspond |
| / | Répertoire racine |
| /bin | Binaires commandes systèmes |
| /boot | Information chargeur démarrage |
| /dev | Périphériques |
| /etc | Configuration machine |
| /home | Répertoires personnels utilisateurs |
| /lib | Bibliothèques et librairies partagées |
| /media, /mnt | Points de montage |
| /opt | Paquetages logiciels complémentaires |
| /root | Répertoire personnel root |
| /slin | Binaires systèmes réservées root |
| /tmp | Fichiers temporaires |
| /usr | Hiérarchie secondaire |
| /var | Données variables |

### Ajouter un utilisateur

Pour diverses raisons, et notamment pour des questions de sécurité, nous vous conseillons de créer des utilisateurs afin d'exécuter vos services avec un autre utilisateur root.

N.B : Sachez que l’utilisateur root est le super-admin de votre machine. Il a tous les droits sur la machine !

Pour ajouter un utilisateur, vous devez utiliser la commande “**adduser <user>**”. Vous pouvez aussi l’ajouter en précisant quelques options :

**adduser <user> -g** : Définir son groupe principal

**adduser <user> -G** : Définir son groupe secondaire

**adduser <user> -M** : Ne pas créer de répertoire personnel pour l’utilisateur

**adduser <user> --home** : Choisir le répertoire personnel de l’utilisateur (par défaut, l’utilisateur va créer un répertoire dans le dossier /home portant le nom de l’utilisateur)

Pour le modifier, vous devez utiliser la commande “**usermod <user> + l’option que vous souhaitez**”.

### Gérer les permissions des fichiers / répertoires

Il existe 3 sortes de droits sur Linux :

* R : La lecture d’un fichier
* W : L’écriture d’un fichier
* X : L’exécution d’un fichier

Pour voir quels sont les droits accordés à un fichier et/ou d’un répertoire vous pouvez faire la commande “**ls -l <nom de votre fichier>** afin d’avoir les droits exacts. Cela donnera quelque chose comme cela :

Ls -l

Cela se lit 3 caractères par caractères, ce qui veut dire pour se fichier :

* Le propriétaire de ce fichier à le droit de lire et d’écrire sur le fichier
* Le groupe propriétaire a le droit de lire le fichier
* Les autres utilisateurs ont aussi le droit de lire le fichier

Pour modifier les droits vous pouvez le faire de deux manières différentes :

Par chiffre :

* R correspond au chiffre 4
* W correspond au chiffre 2
* X correspond au chiffre 1

Si vous souhaitez par exemple que l’utilisateur à tous les droits mais les autres rien vous mettra **“chown 700 <nom du fichier>”.**

Avantages de cette méthode :

* Très pratique quand on veut mettre pleins droits à tout le monde ou enlever tous les droits d’un ou plusieurs membres

Inconvénients de cette méthode :

* Change automatiquement tous les droits. Il faut donc recalculer si on veut laisser les mêmes droits au groupe propriétaire (par exemple).

Par lettre :

* Pour cette méthode “u” correspondra au groupe propriétaire, “g” au groupe propriétaire et “o” pour les autres utilisateurs

De ce fait si on veut juste ajouter tous les droits au groupe propriétaire sans modifier les autres droits nous faisons : “**chmod g+rwx <nom du fichier>”.**

### Vérifier les mises à jour des paquets

Pour vérifier et mettre à jour les paquets npm installé sur notre bot, nous allons faire cette commande :

Apt full upgrade

# Raccourcis clavier pour coder plus vite

Raccourcis les plus répandus sur n’importe quel IDE (VSCode, PhpStorm…).

|  |  |
| --- | --- |
| Raccourci | Utilité |
| CTRL + Z | Pour annuler une modification |
|  | Pour annuler le Ctrl + Z que l’on vient de faire |
| Ctrl + S | Connu de tous, pour sauvegarder |
| Ctrl + (touche espace) | Pour l’auto-prédiction  Pour éviter de faire des copier/coller, surtout quand on n’a pas la souris. |
|  | Pour mettre automatiquement sous forme de commentaires au lieu de le faire manuellement  Idem pour désactiver le commentaire |
|  | ALT + $ : pour refermer les balises, et rendre le fichier plus lisible  ALT + \* : pour ouvrir toutes les balises |
| CTRL + SHIFT +X | Sélectionner la lettre (avec accent) pour la mettre en majuscule |
| CTRL + D | Raccourci pour supprimer/delete |
| CTRL + (touche delete) | Efface paquet par paquet pour ne pas perdre de temps. |
| CTRL + Tabs | Pour insérer des lignes au lieu de faire la touche « entrer » |
| SHIFT + (flèche) | Pour sélectionner les lignes depuis le clavier et non avec la souris. |
| CTRL + W  CTRL + SHIFT + W | Pour fermer le fichier courant  Pour fermer tous les fichiers ouvert |
| CTRL + F | Pour rechercher dans le document |

Balisés personnalisées à peut-être intégrer soi-même dans l’éditeur.

Double cliquer sur la variable pour la sélectionner directement, au lieu de sélectionner avec la souris.

Triple clic pour sélectionner la ligne directement.

# Lignes pour Compozer

|  |  |
| --- | --- |
| **Commande** | **Description** |
| $composer init | Initialisation de composer dans notre projet, et créons en même temps le fichier composer.json en renseignant des informations qui va nous être demander par composer:   * Le nom du projet que vous pouvez laisser par défaut * La description du projet * L'auteur * Stabilité minimale, il va permettre de filtrer les librairies par rapport à la valeur qui vous lui attribuez. J'y ai mis stable, comme ça tous les packages que je vais utiliser doivent être à la version stable. Rendez-vous [ici](https://getcomposer.org/doc/04-schema.md#minimum-stability) pour plus d'informations * Type du package : nous nous développons un projet, donc on écrit project * License : la licence va définir comment les autres personnes pourront utiliser votre projet * Les deux dernières questions répondez non. |
| $composer require … | Installer le package que l’on souhaite, donc $composer require + le nom du package  Annuaire de package [www.packagist.org](http://www.packagist.org) |
| $composer update | Mettre à jour notre fichier dans le package. |
| Numérotation des versions :   |  |  |  | | --- | --- | --- | | 1. | 1. | 1 | | Version majeure | Version mineure | Petit changement | | |

# Lignes de commandes Symfony

* Créer un nouveau projet :

**composer create-project symfony/website-skeleton my\_project\_name**

(my\_project\_name à remplacer avec le nom de son projet)

* Afficher le projet en local : **php –S localhost:8080 –t public**
* Créer un controller : **php bin/console make:controller**
* créer une entité : **php bin/console make:entity**
* créer une base de données **php bin/console doctrine:database:create**
* appliquer les changements (mise à jour des fichiers) **: php bin/console make:migration**
* appliquer la migration (avec validation) **: php bin/console doctrine:migrations:migrate**

## User et authentification

* créer un utilisateur : **php bin/console make:user**

(Yes -- Enter x2 -- Email OU pseudo – Enter)

* créer une authentification : **php bin/console make:auth** -- 1 -- Nom de la classe (avec AUTHENTICATOR à la fin) --Enter --Yes
* créer un formulaire d'inscription : **php bin/console make:registration-form** (Y / Y / adresse email / name for email / Y)
* commande pour faire un CRUD : **php bin/console make:crud**

## Configuration de twig avec bootstrap

* dans le fichier config/packages/twig.yaml, mettre dans le fichier cette ligne : **form\_themes: ['bootstrap\_4\_layout.html.twig’]**