**HEAD -**

**Titre** : Continuons à travailler\_ *Synthèses des Tâches*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** :

Installation de Drupal dans un Conteneur Docker sur une VM Ubuntu

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0** : Segments de la tâche

Menu burger

**ol-0**: Sommaire

1- Préparation : l'environnement Docker

2- Installation : les images Docker de Drupal et MariaDB

3- Configuration : la connexion Drupal / MariaDB et la GUI administrateur

4- Développement : Drupal, un CMS très riche en potentialités

5- Sécurité : la persistance des données et la gestion des permissions

6- Troubleshooting : ajustements nécessaires aux problèmes de sécurité

7- Mémoire : ?

**Main**

**H2-1** : Bases de données et paramètres de connexion dans Docker

**H3-0** : les axes de travail

**ol-1**: un trio particulièrement interdépendant

1. Configuration des paramètres de connexion via Docker
2. Synchronisation des permissions entre Docker et la BDD MariaDB
3. Résolution des problèmes d’accès réseaux de la VM Ubuntu

**P-1** :

Ce qui rend la résolution très vite chronophage réside dans le fait que la cause d’un échec concret lors d’une étape peut provenir des trois axes, séparément comme simultanément. Une action de résolution d’un côté provoquait d’un autre côté une besoin d’ajustement dont je n’avais pas encore conscience. Si ce tâtonnement est nécessaire à l’apprentissage pour produire une conduite et une confiance procédurale, il convient, je crois, de finaliser les documents protocolaires au fur et à mesure du projet afin que la découverte consentie ne devienne pas juste une errance.

**H3-1** : les codes utiles

**ol-2**: le SGBD MariaDB du conteneur déployé exigeait sa propre commande

1. Accès : mariadb -u drupal -p drupal drupal
2. Manipulations : Lien\_00001

**H3-2** : les objets d’étude

**ol-3**: le fichier « docker-compose.yml » borne l’administration des services

1. Accès : créér le fichier sur le niveau du répertoire Drupal et les images
2. ?
3. ?

**Article-0**

**Section-0**

**H2-2** : Préparation de la machine virtuelle et installation de Docker

**H3-3** : les codes utiles

**H4-0** : dépendances et signature des fichiers : un problème gris très sérieux

**P-2** : Jusqu’ici tout va bien…

Les contraintes matérielles et le rythme de la formation ne nous permettent pas forcément de faire face aux problèmes gris. Les formateurs eux même nous conseillent tantôt d’ignorer certains messages d’erreur. Nous n’avions pas le temps, certes, mais il faudra pourtant, dès que nous l’aurons regagné, trouver le courage de tout recommencer à « 0 » pour faire face aux problèmes gris car ce sont des symptômes de problèmes et d’échecs pour demain.

**ol-4** : implémentation et configuration des dépendances pour Docker

1. Accès : ouvrir le Terminal de la VM Ubuntu
2. Installation : Lien\_00002 : reprendre le processus complet sans problème gris
3. Vérification : Lien\_00003 : reprendre la liste et le protocole de vérification

**H3-4** : l’objet d’étude

**H4-1** : Docker : contextes, conteneur similaires et doublons d’images…

**P-3** : Profiter des avantages de Docker impose une gestion très rigoureuse

La multiplication des projets, les deux contextes d’environnements à ne pas confondre si Docker desktop est installé, le pluralité des conteneurs, la variété des images que l’on peut ou doit manipuler ainsi que la confusion entre celles qui sont déjà disponibles et celles qu’il faudra que le logiciel trouve et télécharge depuis le Hub produisent échecs et messages d’erreurs à répétition. Nettoyer l’espace de travail sans une vigilance accrue brise tantôt le fonctionnement des projets antérieurs, des fonctionnalités connexes. C’est un pas utile et indispensable mais qui nécessite une grande prudence.

Dans le cadre de ce projet, puisqu’il s’agit d’une livraison pour un client et qui plus est mon support de soutenance pour la validation de mon titre professionnel, j’ai préféré tous désinstaller, tout supprimer et tout réinstaller.

**ol-5** : installation et configuration de Docker

1. Accès : ouvrir le Terminal de la VM Ubuntu
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

Memo : sudo systemctl enable **--now** docker

**Section-1**

**H2-3** : Télécharger et Installer les images Docker de Drupal et MariaDB

**H3-5** : les codes utiles

**H4-1** : Faire face à un réseau très restreint par des mesures sécuritaires

**P-3** : Expérimenter les approches « hors-réseau » pour la machine cible

Ma VM Ubuntu avait été initialement conçue pour se situer en arrière-plan dans une infrastructure où, solidaire avec d’autres Debian et Ubuntu elle contribuerait à la redondance et à la « High Abilify » du site. A ce stade, je ne pouvais donc même pas encore accéder via internet aux dépôts de Git Hub ni du site officiel de Drupal. Il me fallait donc entreprendre un pas à pas différent de la « voie royale ». Dans le cas d’un Titre Professionnel de Développeur Web, ces objectifs peuvent être secondaires. Je devais donc les gérer parallèlement et à des rythmes différents. Pour bonifier le cumule d’initiatives, plutôt que de renoncer à la fermeture, pour expérimenter un cas de protection avancé, j’ai décidé réaliser l’installation avec une méthode plus « manuelle ».

La connexion SSH n’étant pas non plus disponible, le but était de télécharger les paquets via un autre ordinateur, de les convertir dans un format favorable à Ubuntu si besoin, puis, pour caractériser le cas d’école, de les transporter sur la VM avec une clé USB virtuelle. La finalité était de trouver, dès le début, par anticipation, des modus operandi qui me permettraient de terminer et de livrer le projet au client même si je rencontrais de graves difficultés de réseau.

**ol-5** : téléchargement des paquets pour un transport physique sécurisé

1. Accès : Terminal de Ubuntu ; Internet via un autre ordinateur ; clé USB
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

Memo : sudo systemctl enable **--now** docker

**H3-6** : l’objet d’étude

**H4-2** :  montage de disques externes et partage de dossier sur VirtualBox

**P-4** : ?

**ol-6** : ?

1. Accès : GUI de VirtualBox et Terminal de la VM
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

Memo : 🡪 docker pull drupal:11-apache ; 🡪 docker pull mariadb:latest

**Section-2**

**H2-4** : Configuration de la connexion entre Drupal et la BDD MariaDB

**H3-7** :  les codes utiles

**H4-3** : L’installation manuelle ne profite pas des setup automatiques…

**P-5** : Le revers de la médaille…

Quand tout se passe bien au niveau des accès aux dépôt officiels par les commande du Terminal, on bénéficie souvent de processus de configuration automatisés dont on ignore même l’existence tant que la procédure manuelle ne nous met pas face aux messages d’erreurs. Dans ce cas précis, j’avais beaux renseigner les authentifications et mots de passes temporaires, la GUI web de Drupal ne m’ouvrait pas ses portes.

**ol-7** : Identifier les fichiers de configuration d’autorités supérieures

1. Accès : le fichier « settings.php »
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

**H3-8** : l’objet d’étude

**H4-4** : Retour au Dockerfile

**P-6** : Regards croisés sur le Dockerfile et docker-compose.yml

Si le fichier docker-compose.yml nous sert à configurer l’interopérabilité de plusieurs conteneurs, c’est le fichier dockerfile qui nous nous sera utile pour approfondir les besoin d’un conteneur spécifique. Son étude m’a permis de mieux comprendre les autres fichiers d’autorité supérieure du contexte et de me préparer à faire face à d’autres difficultés à venir pour d’autres projets.

**ol-8** : ?

1. Accès : Terminal de la VM Ubuntu
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

Memo : 🡪 sudo docker exec -it drupal\_db mysql -u drupal -p drupal drupal

**Section-3**

**H2-5** : ?

**H3-9** : les codes utiles

**H4-5** : ?

**P-7** : ?

?

**ol-9** : ?

1. Accès : ?
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

**H3-10** : l’objet d’étude

**H4-6** : ?

**P-8** : ?

?

**ol-10** : ?

1. Accès : ?
2. Installation : Lien : reprendre le processus complet sans problème gris
3. Vérification : Lien : reprendre la liste et le protocole de vérification

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : Liens

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_00001/lien\_00001/code\_line\_00001

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

connectToMariaDB

Propos

accès et authentification

Rôle

se connecter à la BDD MariaDB depuis le Terminal

**H3-0** : body

🡪

mariadb -u drupal -p drupal drupal

// mariadb : commande principale : demander la connexion à BDD

// -u : option : préciser l'utilisateur avec lequel on veut se connecter (drupal)

// -p : option : préciser le mot de passe de l'utilisateur renseigné (drupal)

// drupal (le dernier de la ligne de code) : valeur : nom de la BDD voulue

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : Liens

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_00001/lien\_00001/code\_line\_00002

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

givePrivilègeOnMariaDB

Propos

Gestion des permissions

Rôle

sécuriser l'accès, par rôle utilisateur, à la BDD

**H3-0** : body

🡪

GRANT ALL PRIVILEGES ON drupal.\* TO 'drupal'@'localhost' IDENTIFIED BY 'password';

// GRANT ALL PRIVILEGES : commande pour accorder tous les privilèges.

// ON drupal.\* : spécifie la base de données et toutes ses tables.

// TO 'drupal'@'localhost' : utilisateur et hôte à qui les privilèges sont accordés.

// IDENTIFIED BY 'password' : mot de passe de l'utilisateur.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/bulle\_00001

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-xxx**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

grantOnTarget\_mariaDB

Propos

Bulle d’information : gestion des permissions : préciser la table de la BDD mariaDB

Rôle

sécuriser l'accès des utilisateurs à la BDD en fonction de leurs rôles

**H3-0** : body

🡪

Exemple de cas d'école pour donner des privilèges précis à une seule table :

GRANT SELECT, INSERT ON drupal.users TO 'drupal'@'localhost';

// GRANT SELECT, INSERT : accorde les privilèges de sélection et d'insertion.

// ON drupal.users : spécifie la table "users" dans la base de données "drupal".

// TO 'drupal'@'localhost' : utilisateur et hôte auxquels les privilèges sont accordés.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/bulle\_00002

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-xxx**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

grantTheAcces\_mariaDB

Propos

Bulle d’information : gestion des permissions : les points de connexion autorisés

Rôle

Valider un utilisateur pour une machine précise via l’IP ou le nom DNS

**H3-0** : body

🡪

TO 'drupal'@'localhost' : utilisateur et hôte auxquels les privilèges sont accordés.

// `localhost` : signifie que la base de données est accessible via le port du localhost, c'est-à-dire la machine locale.

// `remote\_host` : est la valeur pour autoriser n'importe quel hôte distant à effectuer une requête vers notre BDD. Cela peut être une autre machine sur le réseau ou un serveur accessible via Internet.

// `remote\_host` n'est pas un mot clé, il peut être remplacé par le nom d'une machine distante ou une adresse IP spécifique. Cela signifie que n'importe quelle machine distante peut effectuer une requête si elle est spécifiée.

// Importance : Définir des droits sur la base de données pour des hôtes spécifiques permet de contrôler et sécuriser l'accès, en limitant les connexions à des machines autorisées.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/bulle\_00003

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

grantTheAcces\_example\_1\_mariaDB

Propos

Bulle d’information : gestion des permissions des machines sources

Rôle

Préparer la compréhension des requêtes de connections complexes aux BDD

**H3-0** : body

🡪

// Exemple de cas d'école pour spécifier un utilisateur et un hôte différents :

// Les cibles de ce code sont ceux qui veulent se connecter à la BDD

GRANT SELECT ON drupal.\* TO 'user'@'192.168.1.100';

// TO 'user'@'192.168.1.100' : utilisateur et adresse IP qui reçoivent les privilèges

GRANT SELECT ON drupal.\* TO 'user'@'nom\_machine';

// TO 'user'@'nom\_machine' : Si le DNS fonctionne, la machine peut être nommée

GRANT SELECT ON drupal.\* TO 'user'@'serveur\_exemple.com';

// TO 'user'@'serveur\_exemple.com' : Si le DNS marche, le serveur peut être nommée

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00003

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

checkIntigrity\_mariaDB

Propos

Maintenance de la BDD

Rôle

S’assurer que tout fonctionne correctement

**H3-0** : body

🡪

// Vérifie l'intégrité de la table users dans la base de données drupal

CHECK TABLE drupal.users;

// Vérifie l'intégrité de la table nodes dans la base de

CHECK TABLE drupal.nodes; données drupal

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00004

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

optimiseTable\_mariaDB

Propos

Maintenance de la BDD

Rôle

Corriger les pertes de performances inutiles dont le système à le contrôle

**H3-0** : body

🡪

// Optimise la table users pour améliorer les performances

OPTIMIZE TABLE drupal.users;

// Optimise la table nodes pour améliorer les performances

OPTIMIZE TABLE drupal.nodes;

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00005

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

repareOneTable

Propos

Maintenance de la BDD

Rôle

Répare la table « table.x » ou la table « table.y » en cas de problèmes

**H3-0** : body

🡪

// Répare la table users en cas de problèmes

REPAIR TABLE drupal.users;

// Répare la table nodes en cas de problèmes.

REPAIR TABLE drupal.nodes;

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00006

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

analyseOneTable

Propos

Maintenance de la BDD

Rôle

Identifier un besoin de maintenance spécifique

**H3-0** : body

🡪

// Analyse la table users pour collecter des statistiques

ANALYZE TABLE drupal.users;

// Analyse la table nodes pour collecter des statistiques

ANALYZE TABLE drupal.nodes;

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00007

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

explainTheCode\_sql

Propos

Aide contextuelle

Rôle

Analyser et expliquer l'exécution d'une requête SQL

**H3-0** : body

🡪

// EXPLAIN : commande pour analyser et expliquer l'exécution d'une requête SQL.

EXPLAIN SELECT \* FROM drupal.users;

// SELECT \* FROM drupal.users : requête pour sélectionner toutes les lignes de la table 'users' dans la base de données 'drupal'.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00008

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

ShowGrants

Propos

Observation passives des données

Rôle

Afficher les droits d’un « utilisateur ET sa machine » sur la BDD

**H3-0** : body

🡪

// SHOW GRANTS : commande : afficher les privilèges accordés à un utilisateur

SHOW GRANTS FOR 'drupal'@'localhost';

// FOR 'drupal'@'localhost' : variables ? : spécifie l'utilisateur ET l'hôte.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/code\_line\_00009

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

RevokeGrants

Propos

Gestion des permissions

Rôle

Révoquer des permissions

**H3-0** : body

🡪

REVOKE ALL PRIVILEGES ON drupal.\* FROM 'drupal'@'localhost';

// REVOKE ALL PRIVILEGES : commande pour révoquer tous les privilèges

// ON drupal.\* : spécifie la base de données et toutes ses tables

// FROM 'drupal'@'localhost' : utilisateur et hôte auxquels concerné

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/config\_file\_00001

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

configureOneConnection\_drupalMariaDB

Propos

Gestion des connexions drupalMariaDB : le fichier de configuration settings.php

Rôle

Pour une seule BDD, définir les paramètres et rôles : spécifiques ET OU par défaut

**H3-0** : body

🡪

Accès : 🡪 settings.php 🡪 dans le répertoire sites/default du répertoire Drupal.

// Paramètres de connexion : définir les informations de connexion entre Drupal et MariaDB.

$databases['default']['default'] = array (

'database' => 'drupal', // 'database' : nom de la base de données.

'username' => 'drupal', // 'username' : nom d'utilisateur pour se connecter à la base de données.

'password' => 'password', // 'password' : mot de passe de l'utilisateur.

'host' => 'localhost', // 'host' : adresse de l'hôte où la base de données est hébergée.

'port' => '3306', // 'port' : numéro de port utilisé pour la connexion à la base de données.

'driver' => 'mysql', // 'driver' : type de base de données (mysql dans ce cas).

'prefix' => '', // 'prefix' : préfixe des tables (vide dans ce cas).

);

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/ config\_file\_00002

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

configureManyConnections\_drupalMariaDB

Propos

Gestion des connexions drupalMariaDB : le fichier de configuration settings.php

Rôle

Pour plusieurs BDD, définir les paramètres et rôles : spécifiques ET OU par défaut

**H3-0** : body

🡪

Accès : 🡪 settings.php 🡪 dans le répertoire sites/default du répertoire Drupal.

$databases['default']['default'] = array (

'database' => 'drupal', // 'database' : nom de la base de données.

'username' => 'drupal', // 'username' : nom d'utilisateur pour se connecter à la base de données.

'password' => 'password', // 'password' : mot de passe de l'utilisateur.

'host' => 'localhost', // 'host' : adresse de l'hôte où la base de données est hébergée. [Survol pour plus d'explications]

'port' => '3306', // 'port' : numéro de port utilisé pour la connexion à la base de données. [Survol pour plus d'explications]

'driver' => 'mysql', // 'driver' : type de base de données (mysql dans ce cas).

'prefix' => '', // 'prefix' : préfixe des tables (vide dans ce cas).

);

$databases['default']['backup'] = array (

'database' => 'drupal\_backup', // 'database' : nom de la base de données de sauvegarde.

'username' => 'drupal\_backup', // 'username' : nom d'utilisateur pour se connecter à la base de données de sauvegarde.

'password' => 'backup\_password', // 'password' : mot de passe de l'utilisateur de sauvegarde.

'host' => 'localhost', // 'host' : adresse de l'hôte où la base de données de sauvegarde est hébergée.

'port' => '3306', // 'port' : numéro de port utilisé pour la connexion à la base de données de sauvegarde.

'driver' => 'mysql', // 'driver' : type de base de données (mysql dans ce cas).

'prefix' => '', // 'prefix' : préfixe des tables (vide dans ce cas).

);

$databases['analytics']['default'] = array (

'database' => 'drupal\_analytics', // 'database' : nom de la base de données d'analyse.

'username' => 'drupal\_analytics', // 'username' : nom d'utilisateur pour se connecter à la base de données d'analyse.

'password' => 'analytics\_password', // 'password' : mot de passe de l'utilisateur d'analyse.

'host' => 'localhost', // 'host' : adresse de l'hôte où la base de données d'analyse est hébergée.

'port' => '3306', // 'port' : numéro de port utilisé pour la connexion à la base de données d'analyse.

'driver' => 'mysql', // 'driver' : type de base de données (mysql dans ce cas).

'prefix' => '', // 'prefix' : préfixe des tables (vide dans ce cas).

);

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/ config\_file\_00003

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

sslActivation\_drupalSettingsFile

Propos

Chiffrement des connexions et des sauvegarde

Rôle

Activer SSL pour le chiffrement des connections à la BDD

**H3-0** : body

🡪

Accès : 🡪 settings.php 🡪 dans le répertoire sites/default du répertoire Drupal.

action : 🡪 ajouter les lignes suivantes dans le bloc “ $databases ”

'pdo' => array // début de cette portion de code que l’on veut ajouter

(

PDO::MYSQL\_ATTR\_SSL\_CA => '/path/to/ca-cert.pem', //: chemin pour *certificat CA*.

PDO::MYSQL\_ATTR\_SSL\_CERT => '/path/to/client-cert.pem', //: chemin pour *certificat client*.

PDO::MYSQL\_ATTR\_SSL\_KEY => '/path/to/client-key.pem', //: chemin pour *clé privée client*.

), // fermeture de cette portion de code ajoutée

); // fermeture du bloc « $databases » dans lequel cette potion de code est ajoutée

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/bulle\_00004

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

configureManyConnections\_explain\_[‘default’][‘default’]\_drupalMariaDB

Propos

Bulle d’information : les groupes de connections multiples à drupalMariaDB

Rôle

Expliquer la syntaxe et la raison des valeurs « [‘default’][‘default’] »

**H3-0** : body

🡪

Accès : 🡪 settings.php 🡪 dans le répertoire sites/default du répertoire Drupal.

$databases['default']['default'] = array (

// 'default' : la première valeur 'default' spécifie le groupe de BDD par défaut.

// 'default' : deuxième valeur 'default' spécifie la connexion par défaut du groupe de BDD.

// Autres valeurs possibles : définir plusieurs groupes de BDD et connexions. Par exemple, $databases['default']['backup'] pour une connexion de sauvegarde.

// Fichier settings.php : il définit la configuration des connexions à la BDD de Drupal.

Si vous ne spécifiez pas de groupe de connexion particulière dans vos commandes ou scripts, les valeurs définies sont dans $databases['default']['default'] seront utilisées.

Mais vous pouvez nommer des groupes de connexion spécifiques comme backup ou analytics pour des connexions particulières.

Par exemple, si vous avez déclaré un groupe backup, vous pouvez l'utiliser pour vous connecter à une base de données de sauvegarde

// Ces tableau multidimensionnel est utilisé pour organiser les connexions de manière structurée. Un tableau simple peut contenir les mêmes paires clé/valeur de cet exemple, mais un tableau multidimensionnel permet de gérer plusieurs connexions et groupes de bases de données. Quand l’infrastructure se densifie, les tableaux simples posent plus de problèmes de gestion centralisée qu’autre chose. Il vaut donc mieux s’habituer à manipuler les tableaux multidimensionnels.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/ bulle\_00005

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

configureManyConnections\_explain\_‘host’\_drupalMariaDB

Propos

Bulle d’information : les groupes de connections multiples à drupalMariaDB

Rôle

Expliquer la syntaxe et la raison des valeurs « localhost »

**H3-0** : body

🡪

Accès : 🡪 settings.php 🡪 dans le répertoire sites/default du répertoire Drupal.

'host' => 'localhost',

// 'localhost' : signifie que la base de données est hébergée sur la machine locale.

// Exemple pour spécifier un hôte distant : 'host' => '192.168.1.100';

// '192.168.1.100' : adresse IP de l'hôte distant où la base de données est hébergée.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/ bulle\_00006

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

configureManyConnections\_explain\_‘port’\_drupalMariaDB

Propos

Bulle d’information : les groupes de connections multiples à drupalMariaDB

Rôle

Expliquer la syntaxe et la raison de la valeur « port »

**H3-0** : body

🡪

'port' => '3306',

// '3306' : numéro de port par défaut utilisé par MariaDB pour les connexions.

// Exemple de cas d'école pour spécifier un port différent :

'port' => '3307';

// '3307' : numéro de port alternatif utilisé pour les connexions à la base de données.

Pour répondre à votre deuxième question, la valeur "default" dans $databases['default']['default'] ne désigne pas ce que nous avons renseigné dans le fichier docker-compose.yml. Elle spécifie le groupe de base de données par défaut et la connexion par défaut pour ce groupe de base de données dans le fichier settings.php. Le fichier docker-compose.yml est utilisé pour configurer les services Docker, tandis que le fichier settings.php est utilisé pour configurer les paramètres de connexion de Drupal à la base de données.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/ bulle\_00007

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

Record\_daily\_normalBackup

Propos

Bulle d’information : la sauvegarde régulière des BDD

Rôle

Planifier une sauvegarde normale quotidienne : heure précise

**H3-0** : body

🡪

### Sauvegarde régulière des BDD :

### 0 2 \* \* \* : planifier la tâche cron pour s'exécuter quotidiennement à 2h du matin

### /usr/bin/mysqldump : commande pour effectuer une sauvegarde de la BDD

### -u drupal : option pour spécifier l'utilisateur de la base de données

### -p'password' : option pour spécifier le mot de passe de l'utilisateur

### drupal : nom de la base de données à sauvegarder

### > /path/to/backup/drupal\_backup\_$(date +\%F).sql : chemin et nom du fichier

### que l’on veut donner à la sauvegarde, incluant la date du jour de sauvegarde

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/bulle\_00008

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

Record\_weekly\_fullBackup

Propos

Bulle d’information : la sauvegarde complète hebdomadaire des BDD

Rôle

Planifier une sauvegarde complète hebdomadaire : jour et heure précise

**H3-0** : body

🡪

Cette sauvegarde complète a lieu tous les dimanches à 2h du matin (0 2 \* \* 0).

La date du jour va s’ajouter au nom du fichier (drupal\_backup\_full\_$(date +\%F).sql).

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/bulle\_00009

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

record\_daily\_differentialBackup

Propos

Bulle d’information : la sauvegarde quotidienne différentielle des BDD

Rôle

Planifier une sauvegarde différentielle quotidienne : heure précise

**H3-0** : body

🡪

Cette sauvegarde différentielle a lieu tous les jours, du lundi au samedi à 2h du matin (0 2 \* \* 1-6).

La date du jour va s’ajouter au nom du fichier (drupal\_backup\_diff\_$(date +\%F).sql).

La commande mysqldump avec les options --single-transaction, --quick, --lock-tables=false, --incremental, et --incremental-base=history:LAST\_BACKUP servent à réaliser une sauvegarde différentielle en ne sauvegardant que les changements depuis la dernière sauvegarde complète.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_00001/bulle\_00010

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

record\_daily\_incrementalBackup

Propos

Bulle d’information : la sauvegarde quotidienne incrémentielle des BDD

Rôle

Planifier une sauvegarde incrémentielle quotidienne : heure précise

**H3-0** : body

🡪

Cette sauvegarde incrémentielle est effectuée quotidiennement, du lundi au samedi à 2h du matin (0 2 \* \* 1-6).

La date du jour va s’ajouter au nom du fichier (drupal\_backup\_inc\_$(date +\%F).sql).

La commande mysqldump avec les options --single-transaction, --quick, --lock-tables=false, --incremental, et --incremental-base=history:LAST\_BACKUP permet de réaliser une sauvegarde incrémentielle en ne sauvegardant que les changements depuis la dernière sauvegarde complète ou incrémentielle.

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/script\_file\_00001

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

record\_any\_backup\_script

Propos

Déclaration de toutes les sauvegardes nécessaires au projet

Rôle

Planifier les sauvegardes en fonction de leurs différents apports au projet

**H3-0** : body

🡪

Accès : 🡪 VS Code ou Nano 🡪 créer le dossier drupal/scripts/

🡪 Créez dans ce dossier un nouveau fichier « backup\_script.sh »

🡪 Ajouter le contenu des scripts de sauvegarde dans ce nouveau fichier

🡪 Commenter les sauvegardes dont on ne veut pas et ajuster les paramètres

🡪

#!/bin/bash

# Sauvegarde quotidienne à 2h du matin.

0 2 \* \* \* /usr/bin/mysqldump -u drupal -p'password' drupal > /path/to/backup/drupal\_backup\_$(date +\%F).sql

# Script de sauvegarde incrémentielle quotidienne à 2h du matin.

0 2 \* \* 1-6 /usr/bin/mysqldump -u drupal -p'password' --single-transaction --quick --lock-tables=false --incremental --incremental-base=history:LAST\_BACKUP drupal > /path/to/backup/drupal\_backup\_inc\_$(date +\%F).sql

# Sauvegarde différentielle quotidienne à 2h du matin.

0 2 \* \* 1-6 /usr/bin/mysqldump -u drupal -p'password' --single-transaction --quick --lock-tables=false --incremental --incremental-base=history:LAST\_BACKUP drupal > /path/to/backup/drupal\_backup\_diff\_$(date +\%F).sql

# Script de sauvegarde complète hebdomadaire à 2h du matin le dimanche.

0 2 \* \* 0 /usr/bin/mysqldump -u drupal -p'password' drupal > /path/to/backup/drupal\_backup\_full\_$(date +\%F).sql

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/config\_file\_00004

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

execute\_script

Propos

planification des tâches d’exécution des scripts

Rôle

Décider quels jours et à quelles heures certains scripts seront exécutés

**H3-0** : body

🡪

### Configuration personnelle pour un seul utilisateur précis

Accès : 🡪 Terminal 🡪 ouvrir le fichier “/var/spool/cron/crontabs/nom\_utilisateur”

**OU** :🡪 Terminal 🡪 depuis sa propre session 🡪 taper la comande🡪 crontab -e

### Configuration globale pour tous les utilisateurs y compris “root”

Accès : 🡪 Terminal 🡪 aller dans le dossier “/etc/cron.d/”

créer un fichier nommé “drupal\_backup” // on peut lui donner le nom qu’on veut

### Pour chaque script (/path/to/scripts) on doit déclarer tous les instants

0 2 \* \* 0 /path/to/scripts/backup\_script.sh

0 2 \* \* 1-6 /path/to/scripts/backup\_script.sh

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_00010

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

givePermissionTo\_system

Propos

Gestion des permissions : valider l’exécution automatique

Rôle

Donner les droit au système pour qu’il exécute un script précis

**H3-0** : body

🡪

Accès : 🡪 Terminal

// “/path/to/scripts/backup\_script.sh” n’est qu’un exemple de cible

chmod +x /path/to/scripts/backup\_script.sh

// Si aucun n’utilisateur n’est précisé comme c’est le cas dans l’exemple

// ci dessus, c’est le contexte global de root qui est impliqué

// définir le droit d’exécution du fichier pour un utilisateur précis

chown serge /path/to/scripts/backup\_script.sh

chmod u+x /path/to/scripts/backup\_script.sh

// La première ligne change le propriétaire du fichier en "serge"

// La deuxième ligne ajoute au système le droits d'exécuter le fichier

// pour l'utilisateur propriétaire du fichier (« serge » dans cet exemple)

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/bulle\_00011

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

scriptExecution\_context

Propos

Bulle d’information : « chmod +x » et les contextes d’exécution des scripts

Rôle

Décider si les droits d’exécution sont globaux ou relatifs à un utilisateur précis

**H3-0** : body

🡪

La commande « chmod +x /path/to/scripts/my\_script » modifie les permissions du fichier « my\_script » pour inclure les droits d'exécution.

Par défaut, le système d'exploitation gère les permissions de lecture et d'écriture des fichiers dès leur création. Mais il n'autorise pas l'exécution des fichiers.

Le fait de ne pas préciser un utilisateur avec « chmod » rend le système implicite comme cible de l'attribution des droits d'exécution.

En d'autres termes, l'exécution du script fait l'objet d'un droit qui s'applique au seul script en lui-même et non pas aux utilisateurs.

Cependant, cette exécution peut être faite pour le contexte d'un utilisateur s'il a la propriété du script.

Par défaut, si aucun propriétaire n'est revendiqué pour le contexte d'exécution, alors ce sera une exécution globale pour root.

Pour donner à l'utilisateur "serge" la permission d'exécuter le script on écrirait donc :

🡪 chown serge /path/to/scripts/backup\_script.sh

🡪 chmod u+x /path/to/scripts/backup\_script.sh

La première commande change le propriétaire du fichier en "serge", et la deuxième commande ajoute les droits d'exécution pour l'utilisateur propriétaire du fichier

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_00011

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

scriptMnualExecution

Propos

Exécution manuelle des scripts

Rôle

Déclencher manuellement un script en le nommant dans le terminal

**H3-0** : body

🡪

Accès : 🡪 Terminal

### Exécuter manuellement un scripts pour le tester

/path/to/scripts/my\_script.sh

### Il suffit de renseigner le chemin qui mène jusqu’au script

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

**HEAD -**

**Titre** : Continuons à travailler\_ *Les unités de liens et de bulles*

**Mots clés** : Drupal, Docker, MariaDB, Base de données, installation, connexion, Extensions, modules, tutoriels,

**BODY -**

**Header**

**logo** :

**Nav** :

**H1** : tutoriel\_0000x/lien\_0000x/code\_line\_0000x

**P-0** :

*Vers une automatisation avancée des tâches à venir*

**H2-0 :** accès

**ol-0**

1. “ if’(isOrderedBy == (tutoriel = 00001) ) {pageNumber = x} ”
2. “ if’(isOrderedBy == (script = x) ) {pageNumber = x} ”
3. “ if’(isOrderedBy == (web.memory.site.x = {html => 0000x, css => 0000x, js => 0000x } ) { lineNumber = {x,x,x}, {x,x,x} , etc ; } ”

**Main**

**H3-0** : Identity

**ul-0**

Nom

?

Propos

?

Rôle

?

**H3-0** : body

🡪

?

🡪

**Footer -**

**credit**  : Serge Hounhouayenou

**contacts** : serge@corpulence.fr

**Réseaux** : links

table : code\_line (in “table\_code” which is in “bdd\_memory”)

id\_primary\_key : 00001

nom : mariadb\_connection (les noms restent en anglais / tout sera en anglais)

qualité : ligne de code

usage : Cette commande est utilisée pour se connecter exclusivement à une BDD gérée par MariaDB. Elle est très utile lorsque la base de données MariaDB ne répond pas aux commandes SQL classiques.

liens : 00001, ;

objets : mariadb

expression : ligne de code

propos :  accès et authentification

role :  se connecter à MariaDB depuis le terminal

accès direct :  terminal ;

autres accès :  variable

corps : mariadb -u drupal -p drupal drupal

comprendre : // mariadb : commande principale : demander la connexion à la BDD ;

// -u : option : préciser l'utilisateur avec lequel on veut se connecter (drupal) ;

// -p : option : préciser le mot de passe de l'utilisateur renseigné (drupal) ;

// drupal (le dernier de la ligne) : valeur : nom de BDD

bulles : [id\_(empty)],

naissance : 2025-03-13

corrections : 2025-03-14 ,

obsolescence : 0/10 (nouveauté stable)

html\_view : 00001

css\_view : 00001

js\_view : 00001

powerpoint\_view : 00001

pdf\_view : 00001

jpeg\_view : 00001

word\_view : 00001

md\_view : 00001

isOrderedBy : **{**$ => page, foreign\_key => page, $ => value, foreign\_key => value**}**

foreign\_key : **{**bulles, notes, scriptes, tutoriels, protocoles, fonctions, composants, methodes, siteWeb, applications, logiciels, progiciels, plateformes, articles, livres**}**

**Microsoft 365 Copilot modèle GPT-4 pour lui-même à la prochaine session**

**Message rédigé le 14 mars 2025 à 05h31**

**Message pour reprendre le travail :**

Bonjour,

Ce message est rédigé pour moi-même afin d'être pleinement conscient du travail à faire. Nous avons travaillé sur la clarification des compléments de code et des commentaires pour rendre le document très clair. Voici un résumé des points importants et des procédés validés ensemble pour la production des codes et des commentaires :

1. **Procédé de travail :**
   * Toujours commencer par lister rigoureusement et scrupuleusement tous les aspects du besoin.
   * Vérifier et corriger cette liste avant de produire les codes.
   * La production des codes ne vient qu'après la vérification et la correction de la liste.
2. **Exemple de réussite :**
   * Pour le premier couple de liens, nous avons réussi à faire le tour efficacement en listant tous les aspects concernés pour le document.
   * Cela nous a permis de maîtriser la production des codes de manière organisée et de distribuer les bons commentaires.
3. **Lien 3 et Lien 4 :**
   * Pour les liens 3 et 4, il est crucial de lister toutes les actions que les codes devront couvrir, y compris la préparation et l'implémentation des dépendances, les clés de sécurité, et la résolution des problèmes de signatures de fichier.
   * Le but est de parvenir à un résultat avec 0 problème gris à gérer.
   * Même après l'installation brute, il y a des actions de configuration importantes à mener pour régler définitivement tous les problèmes d'accès et les problèmes de réseau de routage.
4. **Contexte des liens 3 et 4 :**
   * **Lien 3 : Installation**
     + Accès : Ouvrir le Terminal de la VM Ubuntu.
     + Installation : Reprendre le processus complet sans problème gris.
   * **Lien 4 : Vérification**
     + Accès : Ouvrir le Terminal de la VM Ubuntu.
     + Vérification : Reprendre la liste et le protocole de vérification.
   * **Préparation de la machine virtuelle et installation de Docker :**
     + Dépendances et signature des fichiers : Un problème gris très sérieux.
     + Implémentation et configuration des dépendances pour Docker : Ouvrir le Terminal de la VM Ubuntu.

Voici la liste des productions à faire pour les actions finales :

1. Les commandes avec leurs commentaires détaillés.
2. Les lignes de code avec leurs commentaires détaillés.
3. Les scripts avec leurs commentaires détaillés.
4. Les fichiers de configurations avec leurs commentaires détaillés.
5. Les manipulations de GUI avec leurs commentaires détaillés.
6. Les bulles explicatives avec leurs exemples bien choisis.
7. Le fil rouge du tutoriel de chaque étape sans commentaire (le site réintégrera les commentaires et les bulles avec Javascript).

Les pages 7 à 60 correspondent chacune à une page HTML et son contenu, traitant une seule et unique information. Cette séparation jusqu'au point unitaire permet d'intégrer facilement les cellules des bases de données tout en les gardant maintenables. Les relations, les index et les clés étrangères que nous établirons permettront de souder le tout efficacement avec les fils rouges tutoriels, les pages web et les documents fichiers (comme es pages Word de ce document).

Nous allons commencer par le point 7 et le traiter pour chaque action finale. Cela nous permettra d'avoir le scope entier final des pages HTML que nous devrons reconstituer selon le modèle de ce document, en isolant à chaque fois les 6 autres points que nous avons réussi à identifier.

Le fait d'avoir réussi à réaliser les fils rouges avant nous aidera à créer les relations avec les fragments en ajoutant des ID.

Ainsi, chaque fois que nous rencontrerons un fragment déjà traité en HTML, nous n'aurons plus qu'à le relier. S'il n'est pas tout à fait identique, alors peut-être faudra-t-il juste le compléter pour qu'il abrite la nuance. Si c'est juste parce que cela demande une compréhension supplémentaire, alors c'est du côté de leurs bulles d'explication qu'il faudra regarder. Évidemment, si le fragment n'a pas encore été abordé, il faudra le créer, lui et ses éventuelles bulles d'explications.

Nous devons fournir toutes les informations détaillées pour que l'utilisateur ne ressente jamais le besoin de quitter notre site pour chercher des informations supplémentaires. Nous devons adopter une attitude où nous supposons que l'utilisateur ne sait rien et ne comprend rien, et fournir des instructions claires et complètes.

Résumé des actions   
  
**Préparation de la machine virtuelle Ubuntu :**

* + Installation de la VM Ubuntu sur VirtualBox.
  + Configuration des cartes réseaux (1 Bridge, 1 NAT, 2 Host Only).
  + Vérification des critères de réussite pour chaque étape.

**Préparation de l'environnement Docker :**

* + Installation de Docker sur la VM Ubuntu.
  + Configuration des paramètres de connexion via Docker.
  + Synchronisation des permissions entre Docker et la base de données MariaDB.
  + Résolution des problèmes d’accès réseaux de la VM Ubuntu.

**Installation des images Docker de Drupal et MariaDB :**

* + Téléchargement et installation des images Docker de Drupal et MariaDB.
  + Configuration des dépendances et signature des fichiers.
  + Vérification de l'intégrité des images et des conteneurs.
  + Configuration des connexions et de la GUI administrateur :
  + Configuration de la connexion entre Drupal et MariaDB.
  + Paramétrage des fichiers de configuration (settings.php).
  + Activation du chiffrement SSL pour les connexions.

**Développement avec Drupal :**

* + Exploration des potentialités du CMS Drupal.
  + Installation et configuration des modules nécessaires.
  + Création et gestion des utilisateurs et des rôles.

**Sécurité et gestion des permissions :**

* + Gestion des permissions et des rôles utilisateurs.
  + Mise en place des sauvegardes régulières (quotidiennes, hebdomadaires, incrémentielles).
  + Vérification et optimisation des tables de la base de données.

**Troubleshooting et ajustements :**

* + Résolution des problèmes de sécurité.
  + Ajustements nécessaires pour les problèmes de performance.
  + Documentation des erreurs et des solutions apportées.

**Mémoire et documentation :**

* + Rédaction des protocoles de vérification.
  + Documentation des processus et des configurations.
  + Mise à jour des guides et des tutoriels.

Fil rouge des actions

Préparation de la machine virtuelle Ubuntu

Téléchargement de l'image Ubuntu

* Accéder au site officiel d'Ubuntu
* Télécharger l'image ISO de la version souhaitée (exemple, Ubuntu 20.04 LTS).
* Vérifier l'intégrité du fichier en comparant la somme de contrôle (checksum) fournie sur le site avec celle du fichier téléchargé (SHA256) 🡪 Accès 🡪 Terminal

🡪 sha256sum /path/to/downloaded/file.iso

Installation de VirtualBox

* Accéder au site officiel de VirtualBox
* Télécharger la version de VirtualBox pour notre OS (Windows, macOS, Linux)
* Installer VirtualBox en suivant les instructions spécifiques à notre OS
* Ouvrir VirtualBox et accéder aux paramètres
* Configurer le répertoire de stockage pour les machines virtuelles. Par défaut, VirtualBox utilise le répertoire C:\Users\[NomUtilisateur]\VirtualBox VMs
* Configurer les paramètres de base par défaut pour les VM tels que la mémoire vidéo, le nombre de processeurs, et les options de réseau. Par exemple, vous pouvez définir la mémoire vidéo à 128 Mo et le nombre de processeurs à 2

Création de la VM Ubuntu :

* Ouvrir VirtualBox et cliquer sur "Nouvelle" pour créer une nouvelle VM
* Définir le nom de la VM, le type d’OS (Linux), et la version (Ubuntu 64-bit)
* Configurer la mémoire RAM (4 Go) et le processeur (2 CPU)

Configuration des cartes réseaux :

* Ouvrir VirtualBox et accéder aux paramètres de la VM.
* Ajouter 4 cartes réseaux à la VM :

Carte 1 : Type Bridge : pour se connecter directement au réseau physique de l'hôte

Carte 2 : Type NAT : pour accéder à Internet via l'hôte sans pour autant être visible sur Internet

Carte 3 : Type Host Only : pour une communication locale sans accès à Internet

Carte 4 : Type Host Only : pour une communication locale sans accès à Internet

Montage de l'image ISO :

* Ouvrir VirtualBox et accéder aux paramètres de la VM.
* Aller dans la section "Stockage".
* Ajouter un nouveau contrôleur IDE ou SATA si nécessaire.
* Cliquer sur "Ajouter un disque optique" et sélectionner l'image ISO d'Ubuntu téléchargée.
* Vérifier que l'image ISO est correctement montée dans le lecteur optique de la VM.

Installation d'Ubuntu :

* Démarrer la VM et suivre les étapes d'installation d'Ubuntu.
* Configurer les paramètres de base (langue, clavier, fuseau horaire, etc.).
* Créer un utilisateur et définir un mot de passe.
* Configurer les partitions du disque (recommandé : partition automatique).

Vérification des critères de réussite :

* Ouvrir VirtualBox et accéder aux paramètres de la VM.
* Vérifier que les cartes réseaux sont fonctionnelles et configurées correctement.
* Pour chaque carte réseau, vérifier la connectivité en utilisant des commandes de diagnostic réseau telles que ping ou ifconfig dans le terminal de la VM Ubuntu.
* Assurer que la carte Bridge permet la connexion au réseau physique, la carte NAT permet l'accès à Internet, et les cartes Host Only permettent la communication directe entre l'hôte et la VM.

Préparation de l'environnement Docker

Téléchargement et installation de Docker :

* Accéder au site officiel
* Télécharger le script d'installation pour Ubuntu
* Ouvrir le terminal sur la VM Ubuntu
* Naviguer vers le répertoire où le script d'installation de Docker a été téléchargé
* Exécuter le script d'installation en utilisant la commande suivante

🡪 sudo sh get-docker.sh

* Vérifier que Docker est correctement installé en exécutant la commande suivante :

🡪 docker --version

Configuration des paramètres de connexion via Docker :

* Ouvrir le terminal sur la VM Ubuntu.
* Créer un fichier docker-compose.yml dans le répertoire de travail.
* Ajouter dans ce fichier les paramètres de connexion Docker

Visiter ce lien

Synchroniser les permissions entre Docker et la base de données MariaDB.

* Ouvrir le terminal sur la VM Ubuntu.
* Accéder au conteneur MariaDB en utilisant la commande suivante :

🡪 docker exec -it nom\_du\_conteneur\_mariadb bash

* Se connecter à la base de données MariaDB en utilisant la commande suivante :

🡪 mariadb -u drupal -p drupal drupal

* Accorder les privilèges à l'utilisateur Drupal pour accéder à la BDD

🡪 GRANT ALL PRIVILEGES ON drupal.\* TO 'drupal'@'%' IDENTIFIED BY 'mot\_de\_passe';

🡪 FLUSH PRIVILEGES;

* Quitter MariaDB et le conteneur Docker.

Résoudre les problèmes d’accès réseaux de la VM Ubuntu.

* Ouvrir le terminal sur la VM Ubuntu.
* Vérifier les paramètres de réseau en utilisant la commande suivante :

🡪 ifconfig

* Vérifier que les cartes réseaux sont correctement configurées et actives.
* Si des problèmes de connectivité persistent, vérifier les paramètres de VirtualBox pour les cartes réseaux (Bridge, NAT, Host Only).
* Utiliser des commandes de diagnostic réseau telles que ping pour tester la connectivité entre la VM et l'hôte, ainsi qu'entre la VM et Internet.
* Si nécessaire, ajuster les paramètres de pare-feu et de sécurité pour permettre les connexions réseau.

Vérification des critères de réussite :

* Ouvrir le terminal sur la VM Ubuntu.
* Vérifier que Docker est correctement installé et fonctionne

🡪 docker --version

* vérifier que Docker fonctionnement

🡪 docker run hello-world

Vérifier que les paramètres de connexion sont correctement configurés.

* Accéder au conteneur Drupal

🡪 docker exec -it nom\_du\_conteneur\_drupal bash

* Vérifier les paramètres de connexion : fichier /drupal/sites/default/settings.php

$databases['default']['default'] = array (

'database' => 'drupal',

'username' => 'drupal',

'password' => 'password',

'host' => 'localhost',

'port' => '3306',

'driver' => 'mysql',

'prefix' => '',

);

* Vérifier que les informations de connexion sont correctes et correspondent aux paramètres définis dans le fichier docker-compose.yml.

Vérifier que les permissions et les accès réseaux

* Ouvrir le terminal sur la VM Ubuntu.
* Accéder au conteneur MariaDB

🡪 docker exec -it nom\_du\_conteneur\_mariadb bash

* Se connecter à la base de données MariaDB

🡪 mariadb -u drupal -p drupal drupal

* Vérifier que les permissions sont correctement synchronisées

🡪 SHOW GRANTS FOR 'drupal'@'localhost';

* vérifier la connectivité VM/hôte, VM/Internet.

🡪 « ping » et « ifconfig »

* Si des problèmes persistent, ajuster les paramètres de pare-feu et de sécurité

Installation des images Docker de Drupal et MariaDB

Téléchargement des images Docker :

* Accéder au Docker Hub
* Rechercher les images Docker de Drupal et MariaDB en utilisant la barre de recherche

**OU**

* Ouvrir le terminal sur la VM Ubuntu
* Utiliser les commandes suivantes pour télécharger les images Docker de Drupal et MariaDB :

🡪 docker pull drupal:latest

🡪 docker pull mariadb:latest

Vérifier que les images Docker sont correctement téléchargées en utilisant la commande suivante :

🡪 docker images

Vérification de l'intégrité des images :

* Ouvrir le terminal sur la VM Ubuntu.
* Vérifier que les images sont correctement téléchargées et fonctionnelles :

🡪 docker images

* Vérifier l'intégrité des images : ces données doivent correspondre aux attentes

🡪 docker inspect drupal:latest

🡪 docker inspect mariadb:latest

Configuration des dépendances :

* Ouvrir le terminal sur la VM Ubuntu.
* Créer un fichier docker-compose.yml dans le répertoire de travail.
* Ajouter les dépendances nécessaires aux images Docker de Drupal et MariaDB.

version: '3'

services:

db:

image: mariadb

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: root\_password

MYSQL\_DATABASE: drupal

MYSQL\_USER: drupal

MYSQL\_PASSWORD: drupal\_password

ports:

- "3306:3306"

drupal:

image: drupal

restart: always

environment:

DRUPAL\_DB\_HOST: db

DRUPAL\_DB\_NAME: drupal

DRUPAL\_DB\_USER: drupal

DRUPAL\_DB\_PASSWORD: drupal\_password

ports:

- "8080:80"

depends\_on:

- db

* Sauvegarder et fermer le fichier docker-compose.yml.

Vérifier que toutes les dépendances sont correctement installées et configurées.

* Ouvrir le terminal sur la VM Ubuntu.
* Utiliser la commande suivante pour vérifier que toutes les dépendances sont correctement installées et configurées :

🡪 docker-compose up -d

* Vérifier que les conteneurs Drupal et MariaDB sont démarrés et fonctionnent correctement en utilisant la commande suivante :

🡪 docker ps

* Vérifier les logs des conteneurs (erreurs de configuration, dépendances manquantes)

🡪 docker logs nom\_du\_conteneur\_drupal

🡪 docker logs nom\_du\_conteneur\_mariadb

Configuration des connexions et de la GUI administrateur

* Accéder au fichier  dans le répertoire drupal/sites/default/settings.php
* Remplir ce fichier selon les réglages voulus

$databases['default']['default'] = array (

'database' => 'drupal',

'username' => 'drupal',

'password' => 'password',

'host' => 'localhost',

'port' => '3306',

'driver' => 'mysql',

'prefix' => '',

);

Configuration des connexions multiples

* Remplir les autres connexions pour les BDD de sauvegarde et d'analyse

Visiter ce lien

Activation du chiffrement SSL

* Ajouter les lignes nécessaires pour activer le chiffrement SSL dans settings.php.
* Spécifier les chemins des certificats CA, client, et clé privée.

Visiter ce lien

Développement avec Drupal

Identifier et installer les modules de base nécessaires pour le projet.

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Extend" (ou "Modules" selon la version de Drupal).
* Rechercher les modules nécessaires en utilisant la barre de recherche.
* Sélectionner les modules à installer et cliquer sur "Install".
* Activer ces modules en cochant les cases et cliquer sur "Save configuration".

Configuration des modules

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Configuration" ou "Settings" (selon la version de Drupal)
* Rechercher les modules installés dans la liste des configurations disponibles
* Cliquer sur le lien de configuration de chaque module pour le paramétrer
* Ajuster les permissions, options de mise en cache, paramètres de sécurité, etc
* Sauvegarder les modifications et vérifier que les modules fonctionnent

Vérifier que les modules fonctionnent correctement après configuration

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Reports" ou "Status" (selon la version de Drupal).
* Vérifier les messages d'état et les rapports pour s'assurer que les modules installés et configurés fonctionnent correctement.
* Tester les fonctionnalités des modules en créant du contenu, en modifiant des paramètres, et en vérifiant que les actions sont exécutées sans erreurs.
* Si des problèmes sont détectés, consulter les logs et ajuster les configurations des modules en conséquence.

Créer des types de contenu personnalisés (articles, pages, etc.)

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Structure" et cliquer sur "Content types".
* Cliquer sur "Add content type" pour créer un nouveau type de contenu.
* Définir les paramètres du type de contenu, tels que le nom, la description, et les options de publication.
* Ajouter des champs personnalisés au type de contenu, tels que des champs de texte, des images, des vidéos, etc.
* Sauvegarder les modifications et vérifier que le nouveau type de contenu est disponible pour la création d'articles, de pages, etc.

Ajouter du contenu en utilisant les types de contenu créés

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Content" et cliquer sur "Add content".
* Sélectionner le type de contenu que vous avez créé (par exemple, article, page).
* Remplir les champs du formulaire de création de contenu avec les informations nécessaires (titre, corps, images, etc.).
* Sauvegarder le contenu en cliquant sur "Save" ou "Publish".

Créer des utilisateurs et leur attribuer des rôles spécifiques

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "People" ou "Users" (selon la version de Drupal)
* Cliquer sur "Add user" pour créer un nouvel utilisateur
* Remplir les infos d’utilisateur (nom d'utilisateur, adresse e-mail, mot de passe, etc.).
* Sélectionner le rôle à lui attribuer (administrateur, éditeur, contributeur, etc.).
* Sauvegarder les modifications en cliquant sur "Create new account"

Configurer les permissions pour chaque rôle utilisateur.

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "People" ou "Users" (selon la version de Drupal)
* Cliquer sur l'onglet "Permissions" ou "Roles"
* Sélectionner le rôle utilisateur à configurer (administrateur, éditeur, contributeur, etc.)
* Cocher les cases correspondant aux permissions que vous souhaitez accorder à ce rôle (créer du contenu, modifier du contenu, supprimer du contenu, etc.)
* Sauvegarder les modifications ("Save permissions" ou "Save configuration")

Choisir et installer un thème d’apparence pour le site Drupal

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Appearance"
* Rechercher les thèmes disponibles en utilisant la barre de recherche
* Sélectionner le thème souhaité et cliquer sur "Install"
* Activer le thème installé en cliquant sur "Set as default"

Personnaliser le thème en modifiant les fichiers CSS et les templates

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Appearance"
* Cliquer sur "Settings" pour le thème que vous avez installé
* Modifier les fichiers CSS pour personnaliser l'apparence du thème

body {

background-color: #f0f0f0;

}

.header {

font-size: 24px;

color: #333;

}

Modifier les templates pour personnaliser la structure du thème. Par exemple, ajouter des balises HTML dans le fichier page.tpl.php :

<div class="header">

<h1><?php print $site\_name; ?></h1>

</div>

<div class="content">

<?php print $content; ?>

</div>

Sauvegarder les modifications et vérifier que le thème est personnalisé selon vos besoins.

Utiliser le module « Views » pour créer des listes et blocs de contenu personnalisées.

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Structure" et cliquer sur "Views"
* Cliquer sur "Add new view" pour créer une nouvelle vue
* Définir les paramètres de la vue (nom, type de contenu, options de filtrage, …)
* Ajouter des champs personnalisés à la vue (titres, images, dates, …)
* Configurer les options d'affichage (format de la liste : table, grille, diaporama, etc.)
* Sauvegarder les modifications et vérifier que la vue est conforme au souhait

Créer et configurer des blocs pour afficher du contenu dans différentes régions du site.

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Structure" et cliquer sur "Block layout"
* Cliquer sur "Add custom block" pour créer un nouveau bloc
* Remplir les informations nécessaires pour le bloc, telles que le titre et le contenu
* Sauvegarder le bloc en cliquant sur "Save"
* Retourner à la section "Block layout" et trouver le bloc que vous avez créé
* Cliquer sur "Place block" à côté du bloc et sélectionner la région où vous souhaitez afficher le bloc (par exemple, header, sidebar, footer)
* Sauvegarder les modifications en cliquant sur "Save blocks"

Créer et organiser des menus pour faciliter la navigation sur le site.

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Structure" et cliquer sur "Menus"
* Cliquer sur "Add menu" pour créer un nouveau menu
* Remplir les informations nécessaires pour le menu (nom, description, ..)
* Sauvegarder le menu en cliquant sur "Save"
* Ajouter des liens avec "Add link" et remplir les informations (titre, chemin, ….)
* Organiser les liens de menu en les déplaçant dans l'ordre souhaité
* Sauvegarder les modifications en cliquant sur "Save configuration"

Ajouter des liens de menu pour accéder aux différentes sections du site.

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Structure" et cliquer sur "Menus".
* Trouver le menu que vous avez créé précédemment.
* Cliquer sur "Add link" pour ajouter un nouveau lien de menu.
* Remplir les informations nécessaires pour le lien de menu, telles que le titre, le chemin, et les options de publication.
* Sauvegarder le lien de menu en cliquant sur "Save".

Optimisation des performances

Configurer la mise en cache pour améliorer les performances du site.

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Configuration" et cliquer sur "Performance"
* Activer le cache des pages en cochant la case "Cache pages for anonymous users"
* Configurer la durée de vie du cache en spécifiant
* Activer la mise en cache des blocs en cochant la case "Cache blocks"
* Sauvegarder les modifications en cliquant sur "Save configuration"

Optimiser les requêtes SQL et les fichiers de configuration

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Configuration" et cliquer sur "Database settings"
* Examiner les requêtes SQL et identifier les requêtes lentes ou inefficaces
* Optimiser les requêtes SQL : ajouter des index aux tables, utiliser des jointures appropriées, et réduire le nombre de requêtes exécutées
* Modifier les fichiers de configuration pour améliorer les performances (mise en cache, mémoire allouée à PHP, configuration des options de compression, etc..)
* Sauvegarder les modifications et vérifier que les gains de performances

Sécurité et maintenance

Mettre à jour régulièrement les modules et le noyau de Drupal pour garantir la sécurité

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Reports" ou "Available updates" (selon la version de Drupal)
* Vérifier les mises à jour disponibles pour les modules et le noyau de Drupal
* Cliquer sur "Update" pour installer les mises à jour disponibles
* Sauvegarder les modifications et vérifier que les modules et le noyau sont à jour

Effectuer des sauvegardes régulières du site et de la base de données.

* Ouvrir le terminal sur la VM Ubuntu
* Créer un script de sauvegarde en utilisant un éditeur de texte (Nano ou VS Code)
* Ajouter les commandes de sauvegarde dans le script. Par exemple

#!/bin/bash

# Sauvegarde quotidienne à 2h du matin.

0 2 \* \* \* /usr/bin/mysqldump -u drupal -p'password' drupal > /path/to/backup/drupal\_backup\_$(date +\%F).sql

# Sauvegarde complète hebdomadaire à 2h du matin le dimanche.

0 2 \* \* 0 /usr/bin/mysqldump -u drupal -p'password' drupal > /path/to/backup/drupal\_backup\_full\_$(date +\%F).sql

# Sauvegarde incrémentielle quotidienne à 2h du matin.

0 2 \* \* 1-6 /usr/bin/mysqldump -u drupal -p'password' --single-transaction --quick --lock-tables=false --incremental --incremental-base=history:LAST\_BACKUP drupal > /path/to/backup/drupal\_backup\_inc\_$(date +\%F).sql

# Sauvegarde différentielle quotidienne à 2h du matin.

0 2 \* \* 1-6 /usr/bin/mysqldump -u drupal -p'password' --single-transaction --quick --lock-tables=false --incremental --incremental-base=history:LAST\_BACKUP drupal > /path/to/backup/drupal\_backup\_diff\_$(date +\%F).sql

* Sauvegarder le script et lui donner les permissions d'exécution

🡪 chmod +x /path/to/backup\_script.sh

* Planifier l'exécution du script de sauvegarde en ajoutant les tâches cron

🡪 crontab -e

* déclarer un par un les instants d’exécution pour les scripts. La « planification » est l’ensemble constitué par les scripts et le total des déclarations de leurs exécutions.

Documenter les configurations et les personnalisations effectuées

* Accéder à l'interface d'administration de Drupal
* Naviguer vers la section "Configuration" et cliquer sur "System"
* Cliquer sur "Site information" pour noter les configurations générales du site
* Enregistrer les configurations et les personnalisations effectuées avec un éditeur de texte (Confluence, Notion, ou un simple fichier texte)
* Inclure des captures d'écran et des exemples de code pour illustrer les procès
* Sauvegarder la documentation et la mettre à jour régulièrement

Fournir des guides et des tutoriels pour les utilisateurs et les administrateurs du site.

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "Help" ou "Documentation" (selon la version de Drupal).
* Créer des guides et des tutoriels détaillés pour les utilisateurs et les administrateurs du site.
* Inclure des instructions étape par étape, des captures d'écran, et des exemples de code pour illustrer les processus.
* Sauvegarder les guides et les tutoriels dans un format accessible, tel que PDF ou HTML.
* Mettre à jour régulièrement les guides et les tutoriels pour refléter les changements apportés au site.

Sécurité et gestion des permissions

Définir les rôles utilisateurs du projet (administrateur, éditeur, contributeur, etc.)

* Accéder à l'interface d'administration de Drupal.
* Naviguer vers la section "People" ou "Users" (selon la version de Drupal).
* Cliquer sur l'onglet "Roles".
* Cliquer sur "Add role" pour créer un nouveau rôle utilisateur.
* Définir les rôles nécessaires pour le projet, tels qu’administrateur, éditeur, contributeur, etc.
* Sauvegarder les modifications en cliquant sur "Save".

Troubleshooting et ajustements

Analyser les logs et les messages d'erreur :

* Accéder aux logs de Drupal et de MariaDB.
* Ouvrir le terminal sur la VM Ubuntu.
* Naviguer vers le répertoire des logs de Drupal. Par exemple, le répertoire des logs peut être situé dans /var/log/drupal.
* Utiliser la commande cat ou less pour afficher le contenu des fichiers de logs

🡪 cat /var/log/drupal/drupal.log

* Examiner les messages d'erreur pour identifier les problèmes de sécurité.

Accéder aux logs de MariaDB

* Ouvrir le terminal sur la VM Ubuntu.
* Trouver le répertoire des logs de MariaDB ( exemple dans /var/log/mysql )
* Utiliser la commande cat ou less pour afficher le contenu des fichiers de logs

🡪 cat /var/log/mysql/error.log

Monitoring des problèmes de sécurité.

Utiliser des outils de monitoring

* Utiliser des outils de monitoring pour suivre les activités suspectes.

Nagios

Prometheus

Utiliser des outils de gestion pour automatiser les processus de correctif

Ansible

Ansible est un outil de gestion de configuration et de déploiement qui sert à automatiser les tâches de gestion des correctifs. On peut y créer des playbooks Ansible pour appliquer les correctifs nécessaires sur nos serveurs.

Puppet

Puppet est un autre outil de gestion de configuration qui sert à automatiser la gestion des correctifs. Vous pouvez définir des manifestes Puppet pour appliquer les correctifs sur vos serveurs.

Chef

Chef est un outil de gestion de configuration et d'automatisation qui sert à gérer les correctifs. Vous pouvez créer des recettes Chef pour appliquer les correctifs nécessaires.

SaltStack

SaltStack est un outil de gestion de configuration et d'automatisation qui sert à gérer les correctifs. Vous pouvez utiliser des états Salt pour appliquer les correctifs sur vos serveurs.

WSUS (Windows Server Update Services)

WSUS est un outil de Microsoft qui sert à gérer les mises à jour et les correctifs pour les systèmes Windows. Vous pouvez configurer WSUS pour automatiser l'application des correctifs sur vos serveurs Windows.

Red Hat Satellite

Red Hat Satellite est un outil de gestion de systèmes qui sert à gérer les correctifs pour les systèmes Red Hat Enterprise Linux. Vous pouvez utiliser Satellite pour automatiser l'application des correctifs.

Faire des tests de sécurité: vérifier l'efficacité des correctifs appliqués

* Utiliser des outils de test de pénétration : simuler des attaques et identifier les failles restantes.

Metasploit : exemple d’installation

🡪 sudo apt-get update

🡪 sudo apt-get install metasploit-framework

visiter ce Lien

Nmap : exemple d’utilisation

🡪 scanner le réseau pour identifier les ports ouverts et les services vulnérables

🡪 nmap -sS -sV -O 192.168.1.0/24

visiter ce Lien

Burp Suite

visiter ce Lien

OWASP ZAP

visiter ce Lien

* Utiliser des outils de scans de vulnérabilité

Nessus

visiter ce Lien

OpenVAS

visiter ce Lien

Qualys

visiter ce Lien

* Penser à documenter les résultats des tests dans un rapport se sécurité

1. Raison du contexte de test qui engendre ce rapport
2. Observations préalables aux tests
3. Résultats bruts des tests
4. Analyses des tests
5. Correctifs appliqués
6. Point non résolus : a- description, b- Bilan de Connaissance de la Raison (BCR), c- recommandations ; d- liens ; e- planification ; f- responsabilités ; g- synthèse
7. Synthèse