

# Choix de solutions Mantis Bug Tracker



# Sommaire

Fonctionnalités du produit : .....	3
Les fonctionnalités de Mantis Bug Tracker :.....	3
Atouts du produit : .....	4
Évolution du produit : .....	4
Architecture du produit :.....	4
Installation du produit sous Debian : .....	5
Configuration du produit :.....	8
Déploiement du produit :.....	8
Exploitation du produit : .....	8
Administration du produit : .....	9

## **Genèse du produit :**

Mantis Bug Tracker, souvent appelé simplement "MantisBT", est un outil open source de gestion des bugs et des incidents.

Mantis Bug Tracker est un outil de suivi de bugs dont les fonctionnalités s'adaptent aussi bien aux petits qu'aux grands projets. Son principe de base est simple : un utilisateur qui détecte un bug, ouvre un ticket, puis ce dernier est suivi par les différents acteurs concernés jusqu'à sa clôture.

Le suivi de d'anomalies offre plusieurs états paramétrables pour le suivi des tickets :

- Nouveau,
- Accepté,
- Confirmé,
- Résolu,
- Fermé.

Lancé en 2000 par Kenzaburo Ito, ce logiciel a été conçu pour offrir une solution légère, simple à utiliser, mais puissante, aux équipes de développement logiciel souhaitant suivre et résoudre les problèmes rencontrés dans leurs projets.

## **Fonctionnalités du produit :**

Les fonctionnalités de Mantis Bug Tracker :

- Gestion des catégories,
- Gestion des utilisateurs,
- Notifications par mail,
- Statistiques,
- Définition des champs de données à saisir,
- Rubrique d'administration pour chaque projet,
- Note des tickets,
- Traçabilité.

## **Atouts du produit :**

- Open Source (gratuit et modifiable),
- Simplicité d'utilisation : Interface intuitive, accessible même aux non-techniciens,
- Modulable (ajout via des plugins),
- Personnalisable,
- Communauté active (si besoin lors de problèmes).

## **Évolution du produit :**

MantisBT a connu plusieurs itérations depuis sa création :

- **2000** : Première version publique,
- **2005** : Introduction des notifications par courriel et de l'authentification avancée,
- **2010** : Support des plugins et amélioration de l'interface utilisateur,
- **2020** : Améliorations en termes de performance, interface modernisée et intégration accrue avec des outils tiers,
- **Aujourd'hui** : Maintien actif par la communauté avec des mises à jour régulières et une compatibilité cloud.

## **Architecture du produit :**

MantisBT repose sur une architecture client-serveur basée sur les technologies suivantes :

- **Serveur web** : Compatible avec Apache, Nginx ou IIS.
- **Base de données** : Supporte MySQL/MariaDB, PostgreSQL et autres bases relationnelles.
- **PHP** : Langage principal pour le backend.
- **HTML/CSS/JavaScript** : Interface utilisateur moderne et réactive.

## **Installation du produit sous Debian :**

**Mise à jour du système :** *apt install et apt upgrade*

## **Installation des dépendances :**

```
root@debian12CLI:~# apt install apache2 php libapache2-mod-php mariadb-server php-mysqli pphp-mbstring php-curl php-gd unzip -y
```

## **Téléchargement de MantisBT :**

```
root@debian12CLI:~# git clone https://github.com/mantisbt/mantistbt.git
Clonage dans 'mantistbt'...
Username for 'https://github.com': Mat45Duv
Password for 'https://Mat45Duv@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal : Échec d'authentification pour 'https://github.com/mantisbt/mantistbt.git/'
root@debian12CLI:~# git clone https://github.com/mantisbt/mantisbt.git
Clonage dans 'mantisbt'...
remote: Enumerating objects: 121536, done.
remote: Counting objects: 100% (960/960), done.
remote: Compressing objects: 100% (426/426), done.
remote: Total 121536 (delta 669), reused 547 (delta 533), pack-reused 120576 (from 4)
Réception d'objets: 100% (121536/121536), 84.46 Mio | 35.52 Mio/s, fait.
Résolution des deltas: 100% (95149/95149), fait.
root@debian12CLI:~# _
```

## **On place les fichiers dans le répertoire :**

```
root@debian12CLI:~# mv mantistbt /var/www/html/mantisbt
```

## **Configuration Apache :**

*nano /etc/apache2/sites-available/mantisbt.conf*

## **Ajouter :**

```
<VirtualHost *:80>
    ServerName exemple.com
    DocumentRoot /var/www/html/mantisbt

    <Directory /var/www/html/mantisbt>
        AllowOverride All
    </Directory>
</VirtualHost>
```

## **Configuration de la base de données :**

```
root@debian12CLI:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

### **Connection au serveur MySQL et création base de données et un utilisateur :**

```
root@debian12CLI:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE mantisbt;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER 'mantisuser'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON mantisbt.* TO 'mantisuser'@localhost';
'> FLUSH PRIVILEGES
'>
```



## Page par défaut d'Apache2 Debian

### Ça marche!

Il s'agit de la page d'accueil par défaut utilisée pour tester le bon fonctionnement du serveur Apache2 après son installation sur les systèmes Debian. Si vous pouvez lire cette page, cela signifie que le serveur HTTP Apache installé sur ce site fonctionne correctement. Vous devez **remplacer ce fichier** (situé dans `/var/www/html/index.html`) avant de continuer à utiliser votre serveur HTTP.

Si vous êtes un utilisateur normal de ce site Web et que vous ne savez pas de quoi parle cette page, cela signifie probablement que le site est actuellement indisponible en raison d'une maintenance. Si le problème persiste, veuillez contacter l'administrateur du site.

### Présentation de la configuration

La configuration par défaut d'Apache2 de Debian est différente de la configuration par défaut en amont et divisée en plusieurs fichiers optimisés pour l'interaction avec les outils Debian. Le système de configuration est **entièrement documenté dans `/usr/share/doc/apache2/README.Debian.gz`**. Reportez-vous à ce document pour la documentation complète. La documentation du serveur web lui-même peut être trouvée en accédant au **manuel** si le paquet `apache2-doc` a été installé sur ce serveur.

La configuration d'une installation de serveur Web Apache2 sur les systèmes Debian est la suivante :

```
/etc/apache2/  
|-- apache2.conf  
|  |-- ports.conf  
|  |-- mods activés  
|  |-- *.charger  
|  |-- *.conf  
|  |-- conf-activé  
|  |-- *.conf  
|  |-- sites activés  
|  |-- *.conf
```

- `apache2.conf` est le fichier de configuration principal. Il rassemble les éléments en incluant tous les fichiers de configuration restants lors du démarrage du serveur Web.
- `ports.conf` est toujours inclus dans le fichier de configuration principal. Il est utilisé pour déterminer les ports d'écoute pour les connexions entrantes, et ce fichier peut être personnalisé à tout moment.
- Les fichiers de configuration dans les répertoires `mods-enabled/`, `conf-enabled/` et `sites-enabled/` contiennent des extraits de configuration particuliers qui gèrent respectivement les modules, les fragments de configuration globale ou les configurations d'hôtes virtuels.
- Ils sont activés par des liens symboliques vers les fichiers de configuration disponibles à partir de leurs homologues `*-available/` respectifs. Ceux-ci doivent être gérés à l'aide de nos assistants `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite` et `a2enconf`, `a2disconf`. Consultez leurs pages de manuel respectives pour des informations détaillées.

Le binaire est appelé `apache2`. En raison de l'utilisation de variables d'environnement, dans la

**Voici lorsque je mets mon adresse IP la page.**

## **Configuration du produit :**

La configuration d'un produit dans Mantis Bug Tracker implique plusieurs étapes pour adapter le système aux besoins spécifiques de l'équipe de développement et du projet :

- **Création d'un projet :** Un projet est créé pour suivre les problèmes spécifiques à un produit ou à une fonctionnalité.
  - Spécification du nom du projet, de la description et des catégories.
  - Définition des rôles et des permissions des utilisateurs pour le projet.
  - Configuration des champs personnalisés si nécessaire, tels que les priorités, les types de bugs ou d'autres critères spécifiques.
- **Paramétrage des notifications :** Les utilisateurs doivent être informés des nouvelles entrées ou modifications dans les tickets.
  - Configuration des alertes par e-mail pour les mises à jour de tickets, les changements de statut, etc.
- **Configuration des workflows et des états des tickets :** Définir les étapes de validation et les transitions entre les différents statuts des bugs, comme "Nouveau", "En cours", "Résolu", "Fermé", etc.

## **Déploiement du produit :**

Une fois le produit (ou projet) configuré, l'étape suivante est le déploiement de Mantis Bug Tracker dans l'environnement de production. Cela inclut :

- **Installation du système :**
  - Télécharger et installer Mantis sur un serveur web avec une base de données compatible (MySQL, PostgreSQL, etc.).
  - Configurer le serveur web pour accéder à l'interface Mantis via un navigateur.
- **Importation des données existantes :** Si le projet est déjà en cours et qu'il existe des données de gestion des bugs dans un autre système, il peut être nécessaire d'importer ces informations dans Mantis.
- **Intégration avec d'autres outils :** Intégrer Mantis avec des systèmes tiers, comme des outils de gestion de versions (Git, SVN), des systèmes de CI/CD, ou des plateformes de messagerie pour les notifications et alertes.

## **Exploitation du produit :**

L'exploitation du produit dans Mantis Bug Tracker se réfère à l'utilisation quotidienne du système pour suivre les problèmes et les projets :



- **Création et gestion des tickets** : Les utilisateurs créent des tickets pour signaler des bogues, des demandes de fonctionnalités ou des tâches à accomplir.
  - Les tickets peuvent être assignés à différents membres de l'équipe pour une résolution.
  - Utilisation des filtres et rapports pour suivre les tickets ouverts, fermés, en attente, etc.
- **Suivi des progrès** : Utilisation des outils de suivi de progression comme les graphiques ou les rapports de statut pour voir l'évolution du projet et identifier les goulots d'étranglement.
- **Collaboration** : Les utilisateurs peuvent commenter les tickets pour discuter des problèmes et des solutions, facilitant la collaboration entre les membres de l'équipe.

### **Administration du produit :**

L'administration du produit se réfère à la gestion continue du système Mantis Bug Tracker pour garantir son bon fonctionnement :

- **Gestion des utilisateurs et permissions** : Ajout, suppression et gestion des rôles et permissions des utilisateurs du système. Par exemple, un administrateur peut définir les accès en fonction des responsabilités, telles que les développeurs, les testeurs ou les gestionnaires de projets.
- **Maintenance du système** :
  - Mise à jour du système Mantis pour inclure les dernières fonctionnalités, correctifs de sécurité et améliorations.
  - Sauvegarde régulière des données pour éviter toute perte.
  - Surveillance des performances du serveur et optimisation pour gérer un grand nombre de tickets.
- **Rapports et analyses** : Génération de rapports personnalisés pour analyser l'activité des tickets, identifier les zones de faiblesse, et surveiller la performance des équipes de développement ou de test.