

Installation d'un environnement de développement standardisé sous Ubuntu

Installer Visual Studio Code avec les extensions requises

Visual Studio Code est l'éditeur de code utilisé par l'équipe de développeurs. Voici comment l'installer sous Ubuntu.

Cette étape est essentielle pour garantir un environnement homogène sur tous les postes de travail. VS Code permet une intégration fluide avec les outils Git, ce qui en fait un choix idéal pour une équipe collaborative.

Ajouter le dépôt APT de Microsoft :

Commence par ajouter le dépôt officiel Microsoft pour installer Visual Studio Code via le gestionnaire de paquets APT.

```
sudo apt update
sudo apt install software-properties-common apt-transport-https -y
wget -q https://packages.microsoft.com/keys/microsoft.asc -O- | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stab
```

Installer Visual Studio Code :

```

client@client-VirtualBox:~$ sudo apt install code
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les NOUVEAUX paquets suivants seront installés :
  code
0 mis à jour, 1 nouvellement installés, 0 à enlever et 15 non mis à jour.
Il est nécessaire de prendre 103 Mo dans les archives.
Après cette opération, 410 Mo d'espace disque supplémentaires seront utilisés.
Réception de :1 https://packages.microsoft.com/repos/vscode stable/main amd64 c
de amd64 1.93.1-1726079302 [103 MB]
103 Mo réceptionnés en 3s (30,5 Mo/s)
Préconfiguration des paquets...
Sélection du paquet code précédemment désélectionné.
(Lecture de la base de données... 227227 fichiers et répertoires déjà installés
)
Préparation du dépaquetage de .../code_1.93.1-1726079302_amd64.deb ...
Dépaquetage de code (1.93.1-1726079302) ...
Paramétrage de code (1.93.1-1726079302) ...
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.27-2
uild1) ...
Paramétrage de apt-transport-https (2.7.14build2) ...
client@client-VirtualBox:~$ wget -q https://packages.microsoft.com/keys/microsof
.asc -O- | sudo apt-key add
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (s
ee apt-key(8)).
OK
client@client-VirtualBox:~$ sudo add-apt-repository "deb [arch=amd64] https://pa
ackages.microsoft.com/repos/vscode stable main"
Dépôt : « deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable ma
in »
Description :
  package for vscode: stable components: main

```

Puis vérifier l'installation de Visual Studio Code.

Installer l'extension GitHub : Pull Requests and Issues (via le terminal):

```

client@client-VirtualBox:~$ code --install-extension GitHub.vscode-pull-request-
github
Installing extensions...
Installing extension 'github.vscode-pull-request-github'...
Extension 'github.vscode-pull-request-github' v0.96.0 was successfully installed
.
client@client-VirtualBox:~$ █

```

Cette commande installe l'extension GitHub pour gérer les pull requests et issues directement depuis Visual Studio Code.

Avantages de GitHub

Large communauté et support :

GitHub est l'une des plateformes de gestion de version les plus populaires au monde, avec une immense communauté. Cela facilite l'accès à des ressources, des exemples de code, des projets open-source, et des tutoriels.

La documentation de GitHub est également très complète et régulièrement mise à jour.

Intégration native avec Visual Studio Code :

GitHub dispose d'une extension officielle pour Visual Studio Code (GitHub Pull Requests and Issues), ce qui permet une gestion fluide des pull requests et des issues directement depuis l'éditeur. Cela simplifie la collaboration entre les développeurs.

Le flux de travail entre Visual Studio Code et GitHub est bien intégré, permettant une expérience utilisateur optimisée.

CI/CD intégrée avec GitHub Actions :

GitHub propose GitHub Actions, un service de Continuous Integration/Continuous Deployment (CI/CD) qui permet de créer des workflows d'automatisation (tests, déploiements, etc.) directement dans le dépôt. Cela pourrait être utile pour automatiser les tests fonctionnels sur ce projet.

Sécurité renforcée :

GitHub offre des outils de sécurité comme Dependabot, qui vérifie les dépendances des projets pour signaler les vulnérabilités de sécurité.

GitHub propose aussi la gestion des permissions fines sur les dépôts, ce qui permet un contrôle strict des accès.

Inconvénients de GitHub:

Limitations des dépôts privés dans la version gratuite :

Bien que GitHub permette d'utiliser des dépôts privés gratuitement pour les petites équipes, il y a des limitations dans la version gratuite. Pour des fonctionnalités avancées (comme des outils de gestion de projet plus poussés, des quotas CI/CD, ou des statistiques d'utilisation avancées), il est nécessaire d'utiliser GitHub Teams ou GitHub Enterprise, qui sont payants.

Dépendance à une plateforme externe :

Si l'entreprise préfère une solution plus contrôlée en interne (comme GitLab autohébergé), GitHub peut ne pas être idéal, car il est principalement basé sur le cloud. Bien qu'il existe GitHub Enterprise, cela nécessite une gestion supplémentaire et des coûts supplémentaires pour l'hébergement en local.

Fonctionnalités CI/CD limitées dans la version gratuite :

Bien que GitHub Actions soit un excellent outil, la version gratuite a des limites sur le nombre de minutes d'exécution de workflows par mois, en particulier pour les projets privés. Cela peut entraîner des coûts supplémentaires si l'équipe développe intensément avec de nombreux tests automatisés.

Hébergement Cloud soumis aux lois américaines :

GitHub est une entreprise américaine et donc soumise aux lois et réglementations des États-Unis. Cela peut poser des problèmes de souveraineté des données si Corporate LM a des exigences strictes sur le stockage des données dans certaines juridictions (comme la France ou l'Union Européenne).

Raison :

GitHub est préférable à GitLab pour ce projet en raison de son intégration fluide avec Visual Studio Code, simplifiant le travail des développeurs. Il offre également une large communauté de support, des fonctionnalités de collaboration avancées, et GitHub Actions pour automatiser les tests. Pour une équipe cherchant simplicité, productivité, et compatibilité immédiate, GitHub est le choix optimal.

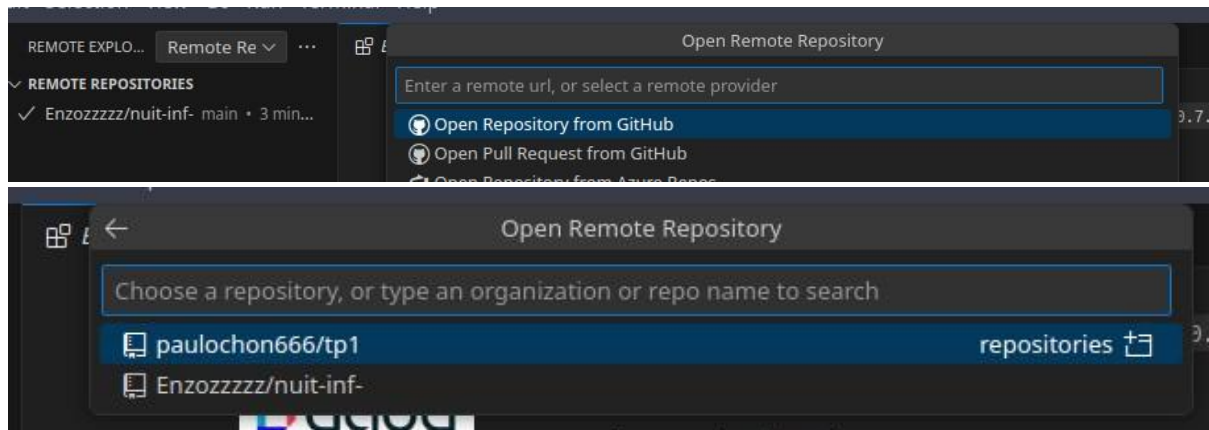
Règles de sécurité

Configurer un pare-feu (Firewall)

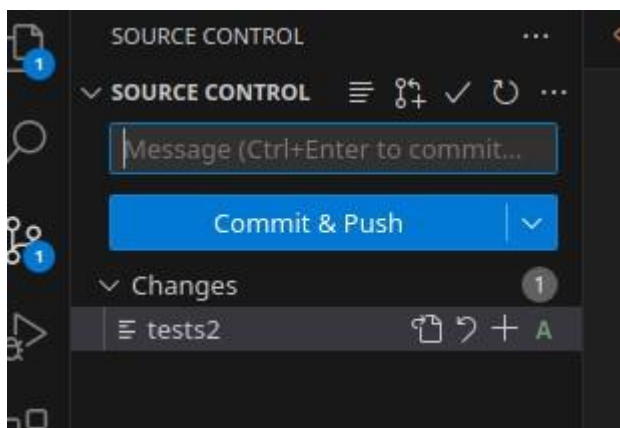
Utilise UFW (Uncomplicated Firewall) pour gérer le pare-feu et autoriser seulement le trafic nécessaire.

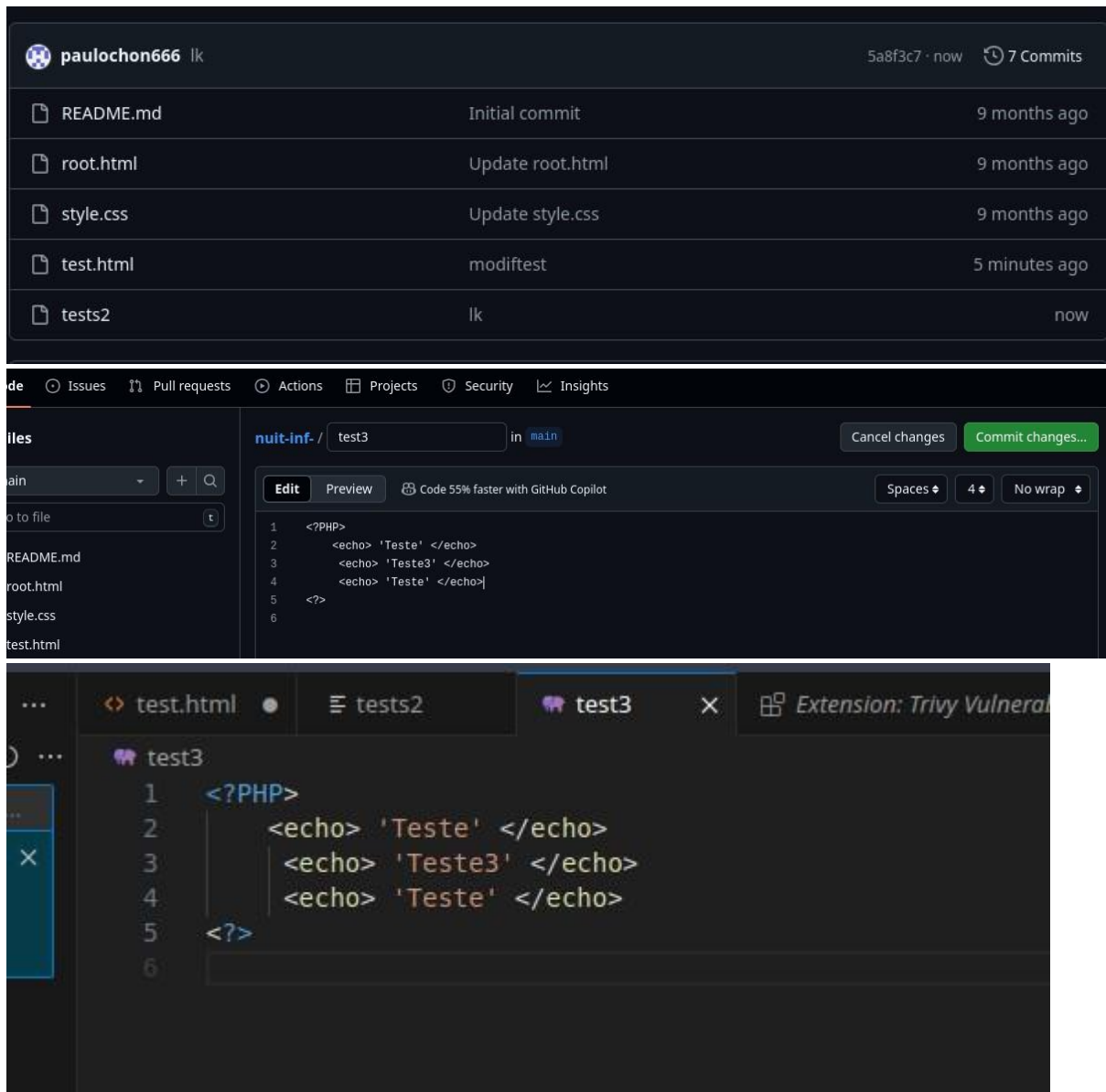
```
client@client-VirtualBox:~$ sudo ufw enable
Le pare-feu est actif et lancé au démarrage du système
client@client-VirtualBox:~$ sudo ufw allow ssh
La règle a été ajoutée
La règle a été ajoutée (v6)
client@client-VirtualBox:~$ sudo ufw allow http
La règle a été ajoutée
La règle a été ajoutée (v6)
client@client-VirtualBox:~$ sudo ufw allow https
La règle a été ajoutée
La règle a été ajoutée (v6)
client@client-VirtualBox:~$
```

Synchronisé le projet GitHub:



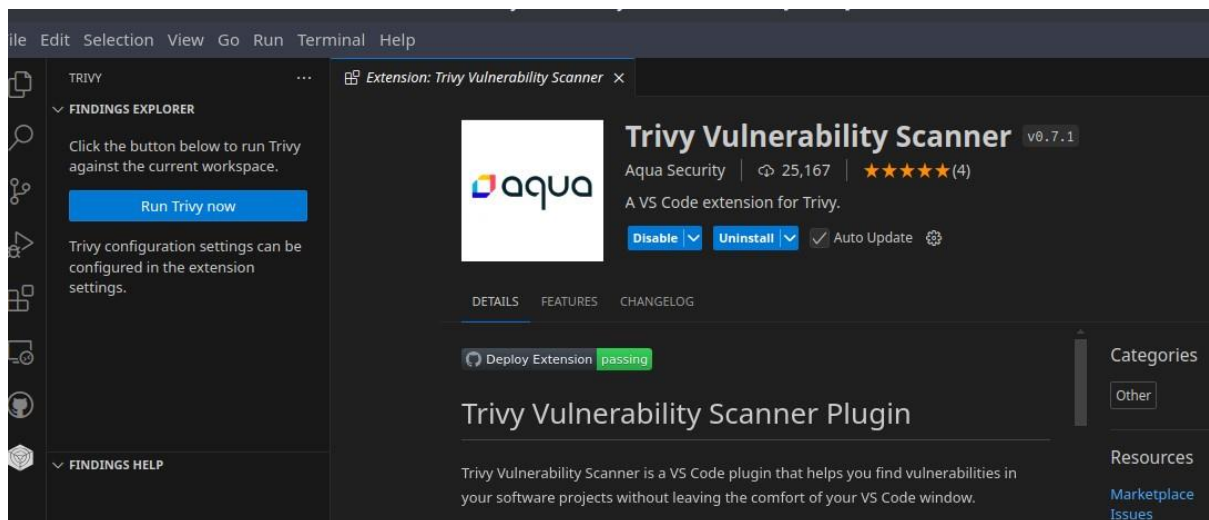
Ensuite, effectuez un test du push pour vérifier si les modifications effectuées localement sont bien synchronisées avec le dépôt distant sur GitHub. Cela consiste à pousser vos **commits** locaux en utilisant la commande **git push** afin de vous assurer que le dépôt distant contient bien les mêmes changements que votre copie locale.





Installation de Tryvi:

Trivy est un outil open-source développé par Aqua Security qui permet de scanner les images Docker, les systèmes de fichiers et les projets pour identifier les vulnérabilités de sécurité, les configurations faibles et les secrets. Il peut être intégré dans Visual Studio Code (VS Code) pour analyser le code et les conteneurs directement dans l'IDE.



Paul Khella

Lénnaïk, Desir