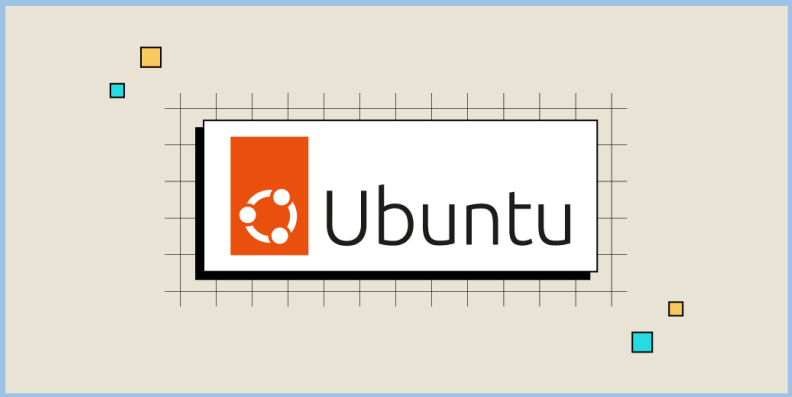
SISR

Installation d'environnement



Cette documentation a pour objectif de détailler la mise en place d'un environnement de travail sur Ubuntu, en précisant les règles de sécurité à respecter lors de l'installation d'un poste de travail. Elle inclut également les instructions pour l'installation de Visual Studio Code, ainsi que des extensions GitLab Workflow et GitHub Pull Requests and Issues.

Saint Paul Bourdon Blanc

Table des matières

Saint Paul Bourdon Blanc	**Ubuntu**	03 / 02 /2024 Version : 1

OBJECTIF : Cette section de la procédure vise à détailler la mise en place de Ubuntu.

MODE OPÉRATOIRE :

Intro

En entreprise, la version Ubuntu LTS (Long Term Support) est l'une des meilleures options pour plusieurs raisons :

- Les versions LTS d'Ubuntu bénéficient de 5 ans de support, incluant les mises à jour de sécurité, les correctifs de bogues et la maintenance. Cela permet aux entreprises d'utiliser une version stable sans se soucier de mises à jour majeures ou de changements incompatibles durant cette période.

- Pour de nombreuses entreprises, la sécurité et la stabilité sont primordiales. Le support prolongé assure le bon fonctionnement des systèmes en toute sécurité, sans avoir à appliquer des mises à jour fréquentes ou risquées.

- Les versions LTS sont conçues pour offrir une stabilité maximale. Elles sont moins sujettes aux modifications radicales ou aux fonctionnalités expérimentales que les versions non LTS.

- Les fonctionnalités et les logiciels intégrés dans une version LTS sont sélectionnés avec soin et largement testés avant leur déploiement.

Ces versions sont également compatibles avec de nombreux logiciels tiers et bénéficient de mises à jour régulières.

Installation et configuration

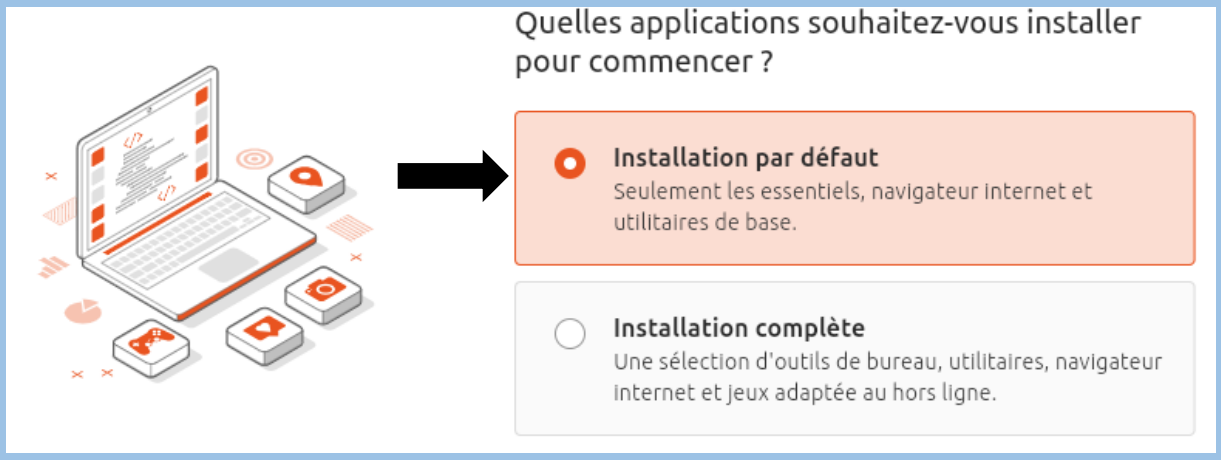
Pour commencer pour pouvoir installer Ubuntu, aller sur le site :

➔ `https://www.ubuntu-fr.org/download/`

L'ors de l'installation d'Ubuntu sélectionner :

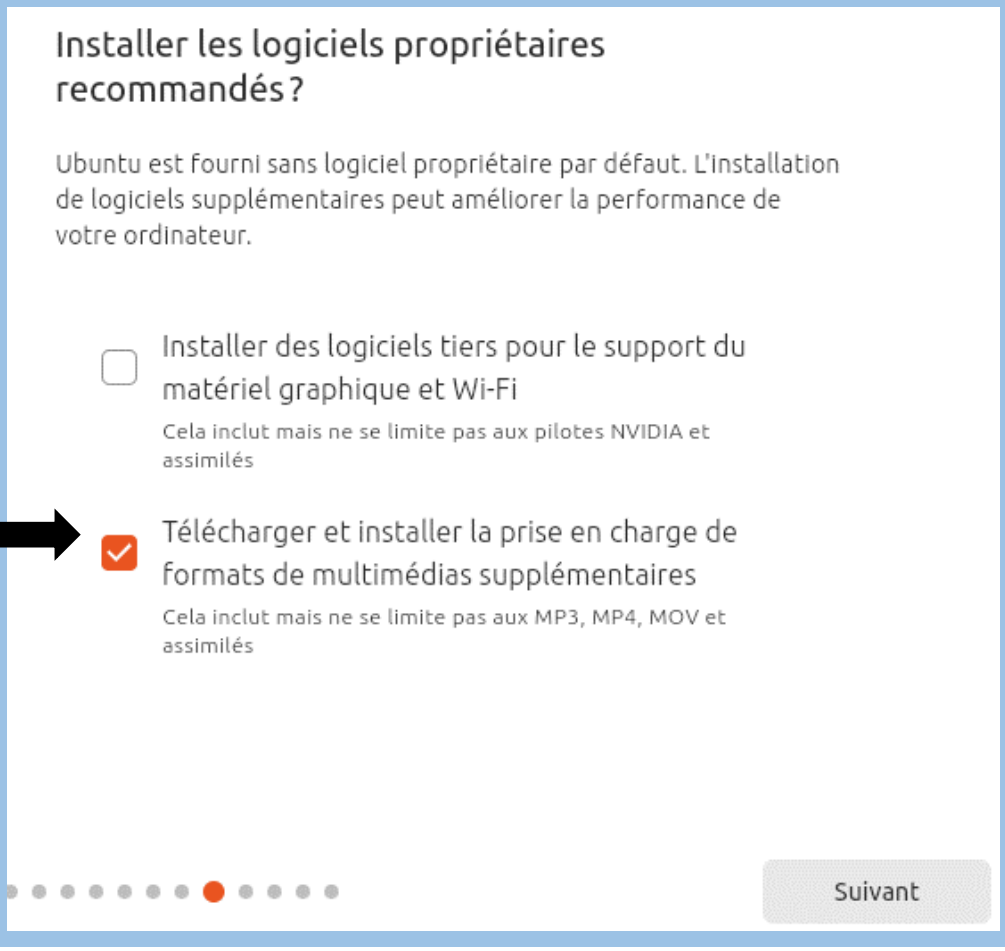
➔

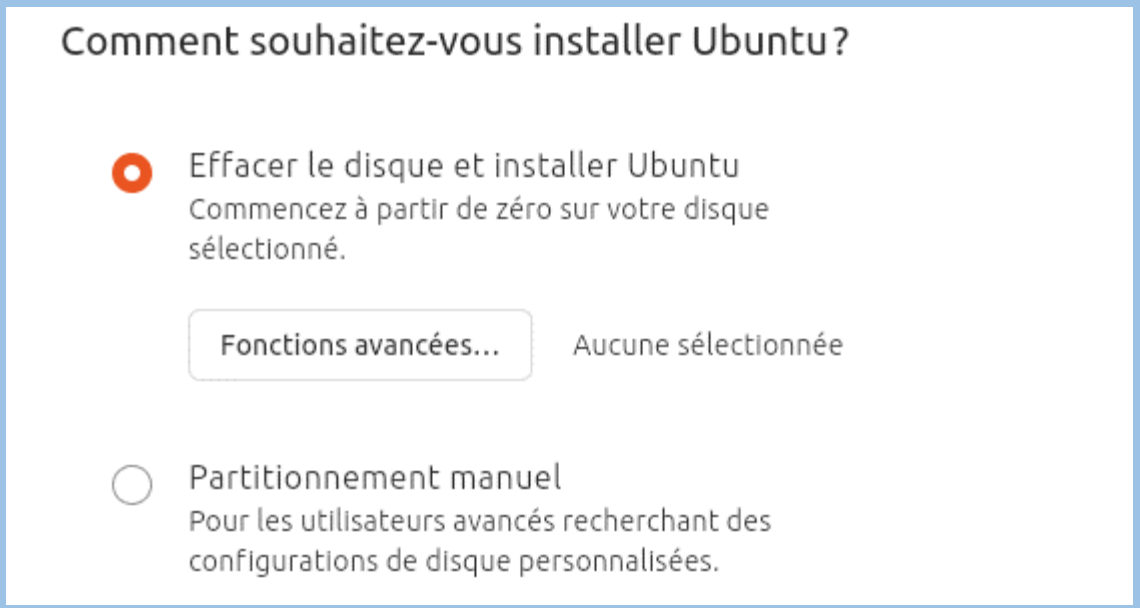
➔

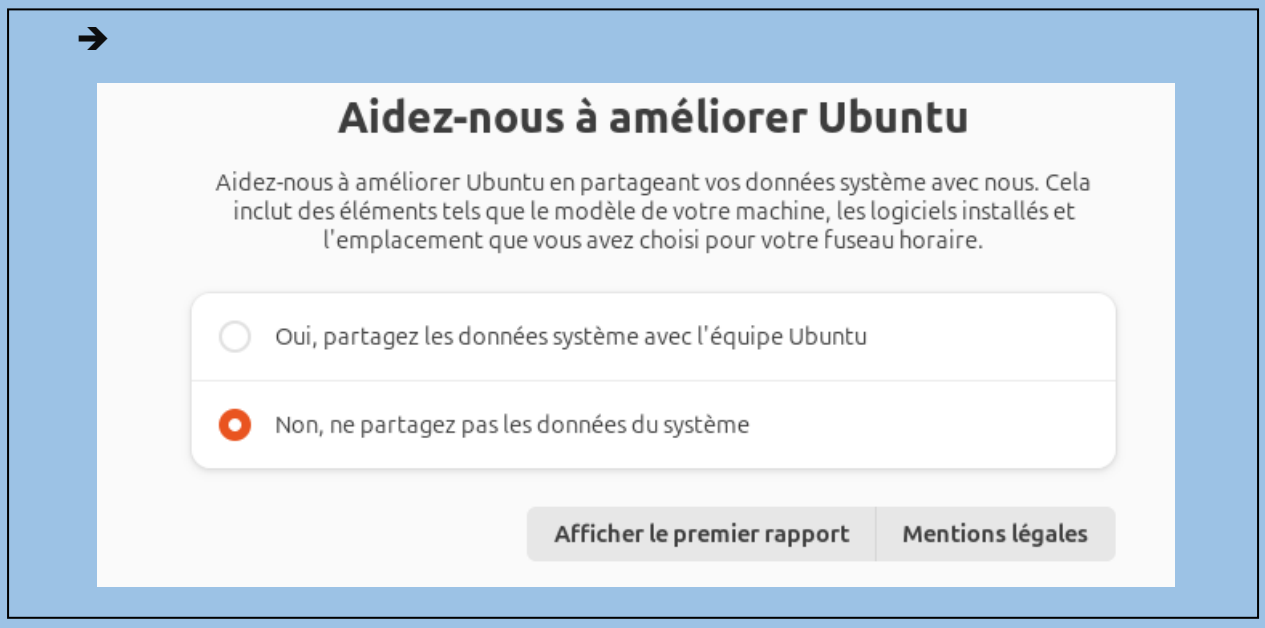


Lors de l'installation des logiciels propriétaires recommandés, cocher seulement la case permettant de prendre en charge les formats multimédias :

➔






Aidez-nous à améliorer Ubuntu
Aidez-nous à améliorer Ubuntu en partageant vos données système avec nous. Cela inclut des éléments tels que le modèle de votre machine, les logiciels installés et l'emplacement que vous avez choisi pour votre fuseau horaire.
Oui, partagez les données système avec l'équipe Ubuntu
Non, ne partagez pas les données du système
Afficher le premier rapport
Mentions légales

OBJECTIF : Cette section de la procédure vise à détailler la sécurisation du poste client.

MODE OPÉRATOIRE :

Créer un compte administrateur

Tapez la commande suivante pour créer un administarteur:

➔ `sudo adduser nom_utilisateur`

```
sio@sio-VirtualBox:~$ sudo adduser admin
```

Tapez la commande suivante pour ajouter l'utilisateur au groupe `sudo` :

➔ `sudo usermod -aG sudo nom_utilisateur`

```
sio@sio-VirtualBox:~$ sudo usermod -aG sudo admin
```

Vérifier les privilèges de l'utilisateur :

➔ `getent group sudo`

```
sio@sio-VirtualBox:~$ getent group sudo
sudo:x:27:sio,admin
```

Créer un compte utilisateur

Tapez la commande suivante pour créer un nouvel utilisateur :

➔ `sudo adduser nom_utilisateur`

```
sio@sio-VirtualBox:~$ sudo adduser user
```

Interdire la modification du réseau

Configurez-le fichier `/etc/sudoers` pour permettre aux utilisateurs d'exécuter uniquement les commandes dont ils ont besoin, sans pouvoir modifier les paramètres réseau :

➔ `user_name ALL=(ALL) ALL, !/sbin/ifconfig, !/sbin/ip`

```
#Réseaux
user ALL=(ALL)ALL, !/sbin/ifconfig !/sbin/ip
```

Protéger les fichiers de configuration réseau :

➔ `sudo chown root:root /etc/netplan/*.yaml`
➔ `sudo chmod 644 /etc/netplan/*.yaml`

```
sio@sio-VirtualBox:~$ sudo chown root:root /etc/netplan/*.yaml
sio@sio-VirtualBox:~$ sudo chown 644 /etc/netplan/*.yaml
```

Limiter les droits d'administration (sudo)

Les utilisateurs doivent généralement avoir des privilèges `sudo` pour installer des logiciels. Vous pouvez ajuster les permissions `sudo` pour restreindre cette capacité dans le dossier « /etc/sudoers » :

➔ `user ALL=(ALL) ALL, !/usr/bin/apt, !/usr/bin/dpkg`

```
#user
user ALL=(ALL)ALL, !/usr/bin/apt, !/usr/bin/dpkg
```

Vous pouvez restreindre l'accès aux commandes comme `apt`, `apt-get`, et `dpkg` en modifiant leurs permissions. Cependant, cette méthode est assez radicale et peut affecter d'autres aspects du système :

➔ `sudo chmod 700 /usr/bin/apt`
➔ `sudo chmod 700 /usr/bin/apt-get`
➔ `sudo chmod 700 /usr/bin/dpkg`

Comment interdire le sudo à l'utilisateur

Sur Ubuntu, les utilisateurs qui ont des privilèges `sudo` sont généralement membres du groupe `sudo`. Vous pouvez retirer un utilisateur de ce groupe pour révoquer ses privilèges `sudo` :

➔ `sudo deluser username sudo`

```
sio@sio-VirtualBox:~$ sudo deluser user sudo
fatal: L'utilisateur « user » n'est pas membre du groupe « sudo ».
```

Interdire le module les périphériques de stockage USB

Créer un fichier de configuration pour blacklist le module USB **:**

➔ `sudo nano /etc/modprobe.d/blacklist-usb-storage.conf`

```
  GNU nano 7.2        /etc/modprobe.d/blacklist-usb-storage.conf
blacklist usb-storage
```

Vous pouvez empêcher les utilisateurs non-autorisés de monter les périphériques USB en modifiant leurs permissions d'accès. Car Ubuntu monte automatiquement les périphériques USB via l'interface utilisateur (comme GNOME).

Créez un fichier de règle Udev pour empêcher le montage automatique :

➔ `sudo nano /etc/udev/rules.d/99-no-usb.rules`

Ajoutez-y le contenu suivant pour bloquer l'accès aux périphériques de stockage USB :

➔ SUBSYSTEM=="usb", ATTR{authorized}="0"

```
  GNU nano 7.2                                /etc/udev/rules.d/99-no-usb.rules *
SUBSYSTEM=="usb", ATTR{authorized}="0"
```

Rechargez les règles Udev :

➔ `sudo udevadm control --reload-rules`

```
sio@sio-VirtualBox:~$ sudo udevadm control --reload-rules
```

Ajout d'une mot de passe pour le bootloader GRUB

GRUB est le gestionnaire de démarrage du système. Sans mot de passe, quelqu'un ayant un accès physique à la machine peut modifier les paramètres de démarrage.

Protéger l'accès non autorisé à la configuration du démarrage, empêchant des modifications dangereuses du système au démarrage.

Générer le hash du mot de passe :

➔ sudo grub-mkpasswd-pbkdf2 « mot de passe »

Ouvrir le fichier de configuration de GRUB :

➔ sudo nano /etc/grub.d/40_custom

Ajout de lignes pour protéger avec un mot de passe :

➔ set superusers="root"
➔ password_pbkdf2 root « hash du mot de passe »

Mettre à jour GRUB :

➔ sudo update-grub

OBJECTIF : Cette section de la procédure vise à détailler la sécurisation du poste avex appArmor.

MODE OPÉRATOIRE :

Introduction

AppArmor (Application Armor) est un **système de sécurité** intégré dans Ubuntu (et d'autres distributions Linux) qui permet de **restreindre les actions des applications** afin de minimiser les risques d'attaques et de compromissions du système. Il fonctionne en appliquant des **profils de sécurité** qui définissent les ressources auxquelles une application peut accéder, telles que les fichiers, les réseaux ou les capacités système.

Configuration

Tout d'abord, commencez par installer AppArmor en exécutant la commande suivante :

➔ `sudo apt install apparmor apparmor-utils`

Une fois cela fait pour générer un nouveau profil pour une application donnée, utilisez la commande suivante :

➔ `sudo aa-genprof /chemin/vers/lapplication`

Une fois un profil créé, vous pouvez l'éditer manuellement pour affiner les règles et les permissions.

Les profils AppArmor sont stockés dans le répertoire `/etc/apparmor.d/` :

➔ `sudo nano /etc/apparmor.d/usr.bin.mon_application`

Les abstractions dans AppArmor sont définies dans le répertoire /`etc/apparmor.d/abstractions/` :

➔ `#include <abstractions/base>`

```
#include <tuable/goal>

/usr/bin/discord{
        /etc/** r,
        /var/log/** rw,
}
```

Installation

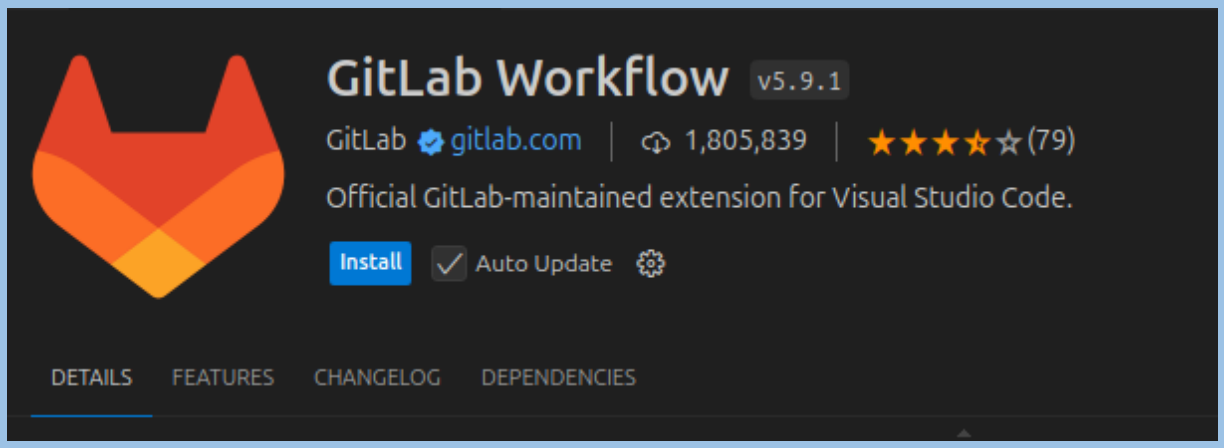
Sous Ubuntu, l'installation peut être effectuée en quelques clics à partir de l'interface graphique, grâce au Centre d'applications du système.

Ouvrez le **Centre d'applications** sur la machine. *L'icône est présente dans le dock, par défaut.*

Recherchez "**visual studio code**" dans le magasin d'applications.

Cliquez sur "**Installer**".

Installation



Configuration

Une fois l'extension installée, vous devrez configurer l'authentification pour que VS Code puisse interagir avec votre compte GitLab.

Générer un token d'accès personnel (PAT) sur GitLab :

Connectez-vous à votre compte GitLab a l'aide d'un navigateur.

Allez dans Paramètres (Settings) > Access Tokens (Tokens d'accès).

Créez un nouveau token en remplissant le champ Nom et en cochant les cases pour les permissions nécessaires (par exemple : api, read_user, read_repository, et write_repository).

Cliquez sur Créer un jeton personnel (Create Personal Token), puis copiez le jeton généré.

Ensuite se connecter à votre compte GitLab a l'aide du jeton via l'extension GitLab Workflow.

Installer git :

➔ sudo apt install git -y

Introduction

Trivy est un outil open-source qui analyse les vulnérabilités dans différents composants logiciels tels que les images Docker, les dépendances de projet, les infrastructures en tant que code (IaC), et bien plus encore. Utiliser Trivy dans Visual Studio Code offre plusieurs avantages :

Sécurité proactive : Il permet d'identifier rapidement les vulnérabilités de sécurité dans vos projets directement depuis votre IDE.

Automatisation : Vous pouvez automatiser le processus de détection des failles dans vos containers Docker, vos fichiers IaC, ou vos dépendances de packages (NPM, Maven, etc.).

Rapidité : Plutôt que d'attendre les tests d'intégration continue, vous pouvez détecter les problèmes directement pendant le développement.

Simplicité : Trivy est léger et facile à configurer, avec une intégration simple dans Visual Studio Code, ce qui le rend accessible aux développeurs sans nécessiter de configurations complexes.

Installation

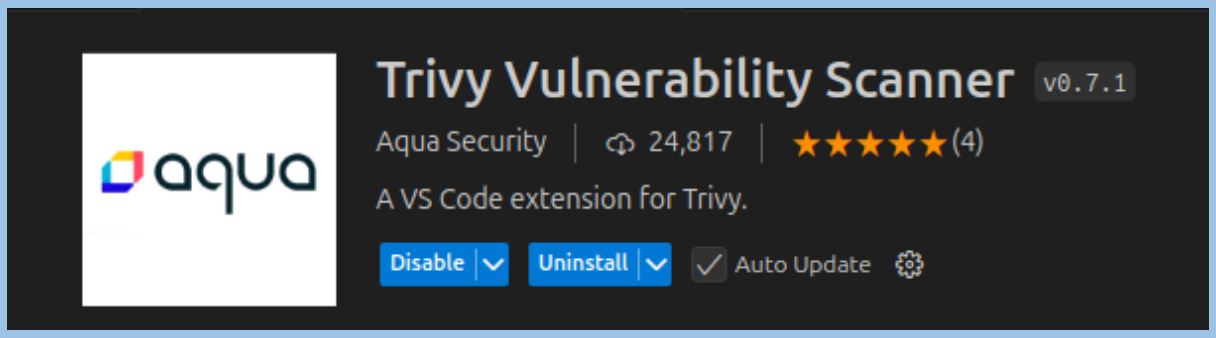
Ajout du dépôt:

- ➔ sudo apt-get install -y wget gnupg lsb-release
- ➔ wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add –

Ajout du dépôt a la liste apt :

- ➔ echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list
- ➔ sudo apt-get update
- ➔ sudo apt-get install trivy

Installer l’extension VS Code :



Utilisation

Scanner un projet

Ouvrez le projet ou le fichier que vous souhaitez analyser (par exemple, un Dockerfile, un fichier de dépendances, ou un fichier de configuration Terraform).

Cliquez droit sur le fichier ou le dossier dans l'explorateur de VS Code.

Sélectionnez l'option "Trivy Scan". Cela va lancer Trivy et afficher un rapport des vulnérabilités directement dans l'IDE.

Analyser les résultats

Une fois l'analyse terminée, Trivy génère un rapport dans le panneau de sortie. Il vous informe des vulnérabilités, de leur sévérité (basse, moyenne, haute, critique), et des suggestions de correction si disponible.

Trivy vous aide à comprendre les risques potentiels et vous oriente vers les actions à prendre pour sécuriser votre code ou vos images Docker.

Saint Paul Bourdon Blanc	**Argumentation**	03 / 02 /2024 Version : 1

Nous avons choisi d'utiliser Ubuntu LTS en entreprise, car, comme mentionné précédemment, elle offre de nombreux avantages en termes de stabilité, sécurité et gestion des coûts. Les versions LTS bénéficient d'un support étendu (5 ans), garantissant des mises à jour régulières et une maintenance fiable, essentielles pour des environnements de production stables. Ubuntu, étant open-source, permet aussi de réduire les frais de licence comparé à des systèmes propriétaires comme Windows Server.

En termes de sécurité, les mises à jour fréquentes protègent contre les vulnérabilités, renforçant la protection des données sensibles. Enfin, Ubuntu dispose d'une vaste communauté, offrant un soutien solide pour répondre aux besoins des entreprises.

Pourquoi choisir GitLab par rapport a GitHub ?

1. CI/CD intégré de manière native

GitLab : Offre un CI/CD intégré et facile à configurer dans toutes les versions, y compris la gratuite.

GitHub : GitHub Actions est utile mais nécessite souvent des configurations complexes et peut engendrer des coûts supplémentaires.

Justification : GitLab est idéal pour automatiser facilement les pipelines sans frais supplémentaires ou outils externes.

2. Cybersécurité et Contrôle des Données

GitLab : Propose l'auto-hébergement, permettant un contrôle total sur les données et la sécurité.

GitHub : Majoritairement cloud avec des fonctionnalités de sécurité avancées dans les plans payants.

Justification : L'auto-hébergement de GitLab permet de répondre aux exigences de sécurité et de conformité, particulièrement important pour les données sensibles.

3. Plateforme DevOps complète

GitLab : Offre une solution DevOps tout-en-un (gestion des versions, CI/CD, sécurité).

GitHub : Nécessite souvent l'intégration d'outils externes pour des fonctionnalités DevOps complètes.

Justification : GitLab centralise tout dans une plateforme unique, réduisant la complexité et les coûts.

4. Gestion des Permissions et Rôles

GitLab : Permet un contrôle granulaire des permissions avec plusieurs niveaux de rôles.

GitHub : Permissions moins flexibles, surtout dans les versions gratuites.

Justification : GitLab facilite la gestion d'équipes aux besoins variés avec un meilleur contrôle des accès.

5. Analyse de Sécurité Intégrée

GitLab : Intègre des outils de sécurité (SAST, DAST) directement dans la plateforme.

GitHub : Dependabot aide pour la sécurité, mais les outils avancés sont limités ou nécessitent des extensions payantes.

Justification : GitLab offre des tests de sécurité intégrés, essentiels pour les équipes soucieuses de la qualité et sécurité du code.

6. Plans Gratuits Plus Riches

GitLab : Propose des fonctionnalités avancées même dans les versions gratuites, notamment pour CI/CD et la sécurité.

GitHub : Certaines fonctionnalités importantes, comme CI/CD étendu et la sécurité avancée, sont réservées aux plans payants.

Justification : GitLab offre plus de fonctionnalités dans son plan gratuit, idéal pour les équipes cherchant à optimiser sans coûts supplémentaires.

7. Conclusion

GitLab est un meilleur choix que GitHub si vous cherchez une solution tout-en-un avec un CI/CD intégré, des fonctionnalités DevOps complètes, et une cybersécurité avancée. L'auto-hébergement et la gestion granulaire des rôles renforcent son attractivité pour les équipes qui ont besoin de flexibilité et de contrôle total sur leurs données.

Editée par	Cylian MENAGE-BIMBENET et Tom COELHO	
Révisée par :	Cylian MENAGE-BIMBENET et Tom COELHO	
Suivie par :	Cylian MENAGE-BIMBENET et Tom COELHO	
Validée par :	Cylian MENAGE-BIMBENET et Tom COELHO	
Date : 3 / 01 / 2023	Saint Paul Bourdon Blanc	Version : 1