Scan réseau avec Ansible et insertion dans Zabbix

Surveiller le réseau et avoir une visibilité

Edouard Bastenier

16/10/2024

CARRÉ BLANC

CARRE BLANC DISTRIBUTION SAS AU CAPITAL DE 1 563 125€ · 10, BOULEVARD DE NANCY 42300 ROANNE
RCS PARIS 400 102 349 · APE-NAF 4641Z · TVA FR56 400 102 349

TABLE DES MATIERES

1. Introduction

Contexte :
Au sein de Carre Blanc, il existe un besoin croissant de disposer d'une vue d'ensemble dynamique et à jour des équipements du réseau local. Actuellement, les informations sur les dispositifs connectés et la topologie réseau doivent être relevées manuellement, ce qui est fastidieux et sujet à des erreurs. L'automatisation de la découverte, de l'inventaire, et de la visualisation des machines et de leurs connexions réseau permettrait de centraliser la supervision et d'assurer une mise à jour en temps réel.

Objectif du document :
Ce document vise à fournir une description technique détaillée du projet d'automatisation réseau pour Carre Blanc. Il décrit les outils, l'architecture, et les étapes de déploiement pour assurer une mise en œuvre réussie de la solution.

2. Environnement utilisé

Machine Hôte :
Le projet sera déployé sur une machine Linux, équipée de Docker pour gérer les différents services dans des conteneurs. Docker permet d'isoler les services, de simplifier la gestion et d'assurer un déploiement reproductible et modulaire des composants. Il me permettra d'héberger des conteneurs pour déployer mon réseau virtuel.

Conteneurs Hébergés :
L'architecture est structurée en 5 conteneurs pour des raisons de performance, de cohérence et de sécurité :

Conteneur 1 : Ansible et Python

Ce conteneur sous sa forme final sera isolé du réseau. Et sera reliée au réseau privé isolé d'internet sur docker.

- Ansible exécute les scripts de scan réseau et gère l'insertion des données dans Zabbix.

- Python exécute Ansible dans un environnement virtuel pour gérer les dépendances.

Conteneur 2 : NMAP
Ce conteneur dédié à NMAP aura accès au réseau local et internet pour effectuer des scans réseau.

Il pourra également communiquer avec les autres conteneurs via le réseau Docker pour permettre à Ansible d'exécuter les commandes NMAP et de récupérer les résultats des scans.

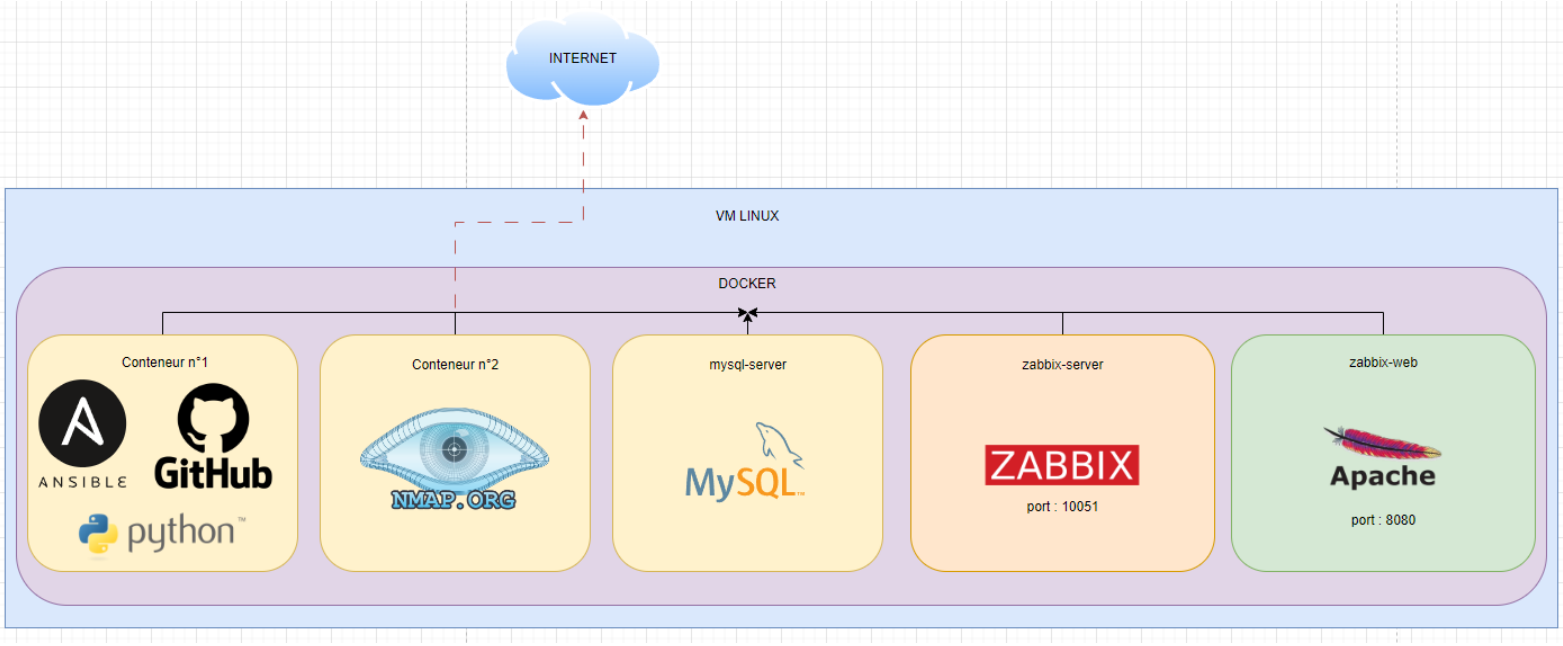
Conteneur 3 : MySQL-server
Ce conteneur dédié à MySQL sera la base de données de Zabbix. Sous sa forme finale, il sera isolé d'internet. Et sera reliée au réseau privé sur docker. Il est installé à partir de l'image du serveur MySQL que nous retrouvons sur le site officiel.

Conteneur 4 : Zabbix-server
Ce conteneur dédié à Zabbix sera le serveur principal de Zabbix. Sous sa forme finale, il sera isolé d'internet. Et sera reliée au réseau privé sur docker. Il est installé à partir de l'image du serveur Zabbix que nous retrouvons sur le site officiel.

Conteneur 5 : Zabbix-web
Ce conteneur dédié à Apache sera la base de données de Zabbix. Sous sa forme finale, il sera isolé d'internet. Et sera reliée au réseau privé sur docker. Il est installé à partir de l'image du serveur Apache dédiée à Zabbix que nous retrouvons sur le site officiel.

Cette topologie permettra d'isoler la partie Ansible et Zabbix sur la machine hôte, puisqu'elles n'ont pas besoin d'avoir accès au réseau local. Et permettra l'accès au réseau local a l'unique conteneur qui en a besoin, le conteneur 3, avec NMAP.

Si on devait l'illustrer voilà à quoi ça ressemblerai :



3. Ansible

3.1 Fonctionnalités principales :

Ansible est un outil d'automatisation de tâches qui fonctionne via SSH pour exécuter des commandes sur des machines cibles, sans avoir besoin d'agent sur celle-ci. Dans ce projet, Ansible :

- Exécutera des scans réseaux avec NMAP pour découvrir les équipements connectés.

- Gèrera la collecte, le traitement, et l'insertion de données dans une base MySQL.

- Automatisera les ajouts et les mises à jour des hôtes dans Zabbix, avec un système historique.

Les playbooks Ansible gèrent l'orchestration des différentes tâches et leur synchronisation.

3.2 Avantages pour l'automatisation :

Ansible est léger et s'intègre bien dans un environnement multi-OS. En automatisant le processus de découverte réseau, il élimine la répétition de tâches manuelles, améliore la précision des données collectées et garantit une exécution rapide des scans et de l'inventaire.

4. Présentation de Zabbix

Zabbix est une solution de supervision open source qui suit l'état des équipements, organise des inventaires, permet de détecter les anomalies, et génère des alertes en cas de problème. De plus il est très intuitif. Dans le cadre de ce projet :

- Zabbix stockera les informations sur chaque équipement découvert dans le réseau.

- Il fournira une visualisation de la topologie réseau, incluant les IP et les noms des machines.

- Zabbix permettra un suivi des performances et de la disponibilité des équipements.

5. Scripts d'Installation et d'Automatisation du déploiement

5.1 Première partie du Script :

Ce script est exécuté sur la machine Linux hôte pour initialiser l'environnement Docker et initialiser les conteneurs :

- Installe Docker sur la machine hôte.
- Monte un dossier contenant des fichiers essentiels (paramètres de configuration, scripts), qui servira de référentiel pour tous les fichiers de configuration modifiés.
- Crée le réseau temporaire et privée.
- Crée les 5 conteneurs Docker avec les configurations adaptées.
- Installe les paquets nécessaire au déploiement du conteneur 1.
- Installe et configure le service SSH et un certificat sur les conteneurs 1 et 2.
- Installe Ansible sur le conteneur 1.
- Installe Python pour supporter Ansible et crée un environnement virtuel dédiée à Ansible.

5.2 Deuxième partie du Script :

Ce script est exécuté à partir d'Ansible pour installer et configurer le reste de l'environnement nécessaire au conteneur 1 :

- Installe les paquets nécessaires du fonctionnement de Git pour le partage et la mise à jour des playbooks et scripts.
- Installe tree pour avoir un affichage de l'arborescence des fichiers au besoin.

- Télécharge le rôle ansible « projet » sur mon GitHub.

5.3 Troisième partie du Script :

Ce script est exécuté à partir d'Ansible pour installer et configurer l'environnement nécessaire au conteneur2 :

- Installe le service nmap.

6. Automatisation des déploiements

Tout le projet se déploiera automatiquement. Grâce au script principal, celui-ci exécutera.

Ces copies sont stockées dans le dossier monté par le premier script pour centraliser la gestion des configurations, faciliter les mises à jour, et garantir la cohérence des services.

7. Déploiement et Tests

Test de Connectivité : Vérification de la communication entre les conteneurs, de l'isolation des conteneurs et de la connexion du container 2 au réseau local.

Test de Découverte et d'Insertion : Vérifie que les playbooks Ansible exécutent les scans correctement et que les données sont bien synchronisées dans Zabbix.

8. Fonctionnement global du projet

Le projet d'automatisation de la découverte et de la surveillance réseau repose sur une architecture modulaire en cinq conteneurs Docker, permettant une gestion sécurisée et centralisée des équipements du réseau de Carre Blanc. Le processus commence avec Ansible, qui utilise un playbook pour exécuter des scans réseau via NMAP (dans le conteneur 2), permettant de découvrir tous les équipements connectés au réseau local paramétré dans le fichier yml. Les résultats des scans sont ensuite récupérés par Ansible, trié et transmis à Zabbix (via MySQL), où les informations sont stockées dans une base de données et affichées sur l'interface web fournie par Apache (dans le conteneur 5 'zabbix-web').

Le fait que les conteneurs où il y a de l'exécution et où est exécuté Zabbix soit isolé du réseau internet, renforce la sécurité des données. Uniquement le conteneur 2 où il y a que Nmap est connecté à internet, ce qu'il fait que la seule machine accessible par internet n'est pas précieuse et ne contient pas de données concrètement.

Ce processus permet d'assurer un inventaire réseau dynamique et à sécuritaire, avec une vue centralisée de l'ensemble des équipements détectés.

9. Bénéfices de l'automatisation

Gain de temps : En automatisant la découverte et l'inventaire réseau, le projet réduit significativement le temps et les efforts de gestion, surtout pour les mises à jour régulières de la topologie.

Amélioration de la réactivité : Grâce à la mise à jour en temps réel, les équipes peuvent repérer rapidement des anomalies, améliorer la sécurité et optimiser la gestion des équipements. La configuration de Zabbix permet également de recevoir des alertes en cas de problèmes.

10. Conclusion

Ce projet d’automatisation de la découverte et de la surveillance réseau permet de répondre aux besoins de visibilité et de gestion dynamique des équipements de Carre Blanc. En intégrant Ansible pour la découverte réseau et Zabbix pour la supervision, cette solution centralise les informations, met à jour les données automatiquement et élimine les tâches répétitives, renforçant ainsi la sécurité et l’efficacité du réseau de Carre Blanc.

De plus, la sécurité du projet a été une priorité tout au long de son développement. En isolant les services dans des conteneurs Docker, chaque composant fonctionne dans un environnement sécurisé et distinct, réduisant les risques d'interférences et d'attaques potentielles. Les communications entre les conteneurs sont contrôlées, avec des restrictions d'accès strictes, ce qui empêche l'accès non autorisé aux données sensibles. L’utilisation de playbooks Ansible permet également de maintenir un environnement cohérent et sécurisé, avec une gestion centralisée des configurations, assurant que toutes les machines et services respectent les mêmes standards de sécurité. Enfin, Zabbix permet une surveillance en temps réel de l'état des équipements, contribuant à la détection précoce d’éventuelles anomalies ou tentatives d'intrusion, garantissant ainsi une réactivité optimale face aux incidents de sécurité.