



كلية العلوم
السملاية - مراكش
FACULTÉ DES SCIENCES
SEMLALIA - MARRAKECH

Université Cadi-Ayyad Marrakech
Faculté des Sciences Semlalia
Département Informatique

RAPPORT DU PROJET DE FIN D'ETUDES

Pour l'obtention du diplôme Licence d'Etudes Fondamentales en :
Sciences Mathématiques et Informatique

Sous le thème :

DEPLOIEMENT D'UNE PLATEFORME WEB DE SUPERVISION RESEAU AVEC NAGIOS

Réalisé par :

Doaa Haji
Oumaima Hamza

Encadré par :

Prof. Mariya Ouaissa

Soutenu le 01/07/2025 devant le jury :

Prof. Mariya Ouaissa

Professeur à FS Semlalia

Prof. Soukaina Mjahed

Professeur à FS Semalalia

Année universitaire : 2024/2025



«قالوا سبحانك لا علم لنا إلا ما
علمتنا إنك أنت العليم الحكيم»

صدق الله العظيم (البقرة ٣٢)

Dédicaces

A nos parents qui nous ont soutenu et encouragé durant nos études.

A nos frères et sœurs qui ont partagé avec nous tous les moments

D'émotion et d'amour lors de notre vie.

A nos encadrants et enseignants qui nous dirigent durant notre

Parcours.

Remerciements

"Au terme de ce travail nous tenons à exprimer notre plus profonde gratitude à notre encadrante Pr. Mariya Ouassa, elle nous a guidé tout au long de notre projet à travers ses conseils et plusieurs directives qu'elle nous a donné.

Nous tenons à dire merci aussi à tous nos professeurs, de nous avoir donné les bagages nécessaires pendant toute cette année universitaire.

Nous remercions aussi les membres du jury pour avoir accepté d'évaluer notre travail.

Nous remercions spécialement nos familles qui sont toujours notre soutien."

Résumé

Dans un environnement où les systèmes d'information occupent une place centrale, la supervision réseau devient un enjeu stratégique pour toute organisation. La disponibilité, la performance et la sécurité des infrastructures informatiques dépendent fortement de la capacité à surveiller en temps réel les équipements, détecter rapidement les anomalies et intervenir efficacement en cas de panne. C'est dans ce contexte que s'inscrit notre projet de fin d'études, dont l'objectif principal est de concevoir et déployer une plateforme de supervision réseau moderne, accessible et fiable, en s'appuyant sur l'outil open source Nagios Core.

Notre démarche a été structurée en deux grandes phases. La première a consisté à installer et configurer l'environnement de supervision basé sur Nagios Core, tout en y intégrant plusieurs outils complémentaires pour étendre ses fonctionnalités comme NagiosQL, NCPA et Thruk.

Dans une seconde phase, nous avons développé une interface web personnalisée, conçue pour permettre aux utilisateurs – même non techniques – de consulter les informations de supervision de manière claire et structurée. Cette interface repose sur les technologies HTML, CSS, JavaScript, PHP, MYSQL et interagit dynamiquement avec les données de Nagios via le socket Livestatus.

Ce travail a également nécessité une modélisation rigoureuse du système à travers des diagrammes UML (cas d'utilisation, classes, séquences), afin de structurer l'architecture logicielle et assurer la cohérence entre les composants. Nous avons accordé une attention particulière à l'ergonomie de l'interface, à la clarté des tableaux de bord, aux outils de filtrage et de recherche, ainsi qu'à la réactivité de l'application.

Ce rapport décrit en détail l'ensemble du processus de réalisation, depuis l'analyse des besoins et la problématique initiale, jusqu'à la mise en œuvre technique et les perspectives d'évolution du système. Il met en lumière les choix technologiques effectués, les difficultés rencontrées, les solutions apportées, ainsi que les limites actuelles du projet, notamment en matière de gestion des utilisateurs, d'interactions actives avec le système et de compatibilité mobile. Des pistes d'amélioration sont proposées pour enrichir la solution dans des versions futures.

Mots-clés : Supervision, SNMP, Nagios, Nagios Core, Nagios QL, Thruk

Abstract

In an environment where information systems play a central role, network monitoring has become a strategic priority for organizations. The availability, performance, and security of IT infrastructures rely heavily on the ability to monitor equipment in real time, quickly detect anomalies, and respond effectively to incidents. It is in this context that our final-year project was developed, with the primary objective of designing and deploying a modern, reliable, and user-friendly network monitoring platform based on the open-source tool Nagios Core.

Our approach was structured in two main phases. The first involved installing and configuring the monitoring environment using Nagios Core, while integrating several complementary tools to extend its functionalities, such as NagiosQL, NCPA, and Thruk.

In the second phase, we developed a customized, intuitive, and responsive web interface designed to allow users – even non-technical ones – to clearly and easily consult monitoring data. This interface was built using HTML, CSS, JavaScript, PHP and MYSQL, and interacts dynamically with Nagios data through the Livestatus socket.

This work also required a rigorous system modeling using UML diagrams (use cases, class and sequence diagrams) to structure the software architecture and ensure consistency among components. Special attention was paid to interface ergonomics, clarity of dashboards, filtering and search tools, as well as the responsiveness of the application.

This report details the entire project development process, from initial needs analysis and problem definition to technical implementation and future enhancement perspectives. It highlights the technological choices made, the challenges faced, the solutions applied, and the current limitations of the system, particularly regarding user management, interactive functionalities, and mobile compatibility. Improvement suggestions are proposed to further evolve the solution in future versions.

Keywords: Monitoring, SNMP, Nagios, Nagios Core, Nagios QL, Thruk,

Table des matières

Liste des figures	ix
Liste des tableaux	xi
Liste des abréviations	xii
Introduction Générale	1
1 Contexte général du projet	2
Introduction.....	2
1.1 Présentation du projet	2
1.2 Problématique	3
1.3 Objectif du projet	3
1.4 Planification	4
1.4.1 Planification du projet	4
1.4.2 Diagramme de Gantt	4
Conclusion	5
2 Concepts de base et état de l'art.....	6
Introduction.....	6
2.1 Les réseaux informatiques.....	6
2.1.1 C'est quoi un réseau	6
2.1.2 Nécessité d'un réseau	6
2.1.3 Types de réseaux.....	6
2.1.4 Equipements Réseaux	7
2.2 Généralités sur la sécurité réseau	8
2.2.1 Objectifs de la sécurité réseau	9
2.2.2 Typologie des menaces réseau	9
2.2.3 Sécurisation d'un réseau informatique	10
2.3 La supervision réseau	10
2.3.1 Définition de la supervision réseau	10
2.3.2 Principe	11
2.3.3 Objectifs de la supervision réseau	11
2.3.4 Les utilisateurs de la supervision réseau	12
2.3.5 Méthodes de supervision	12

2.4	Protocole SNMP.....	13
2.4.1	Définition du protocole SNMP.....	14
2.4.2	Les différentes versions du SNMP.....	14
2.4.3	Architecture et composants	14
2.4.4	Les requêtes SNMP	15
2.4.5	Avantages et limites de SNMP	16
2.5	Les outils de supervision existants	16
2.5.1	Introduction aux outils de supervision réseau	16
2.5.2	Exemples d'outils populaires	17
2.5.3	Benchmarking des solutions.....	19
	Conclusion	21
3	Implémentation de la solution Nagios	22
	Introduction.....	22
3.1	Architecture de Nagios	22
3.2	Présentation de Nagios	24
3.2.1	Les avantages du Nagios.....	25
3.3	L'environnement de Nagios	25
3.3.1	Prérequis matériels et logiciels	25
3.3.2	Installation de Nagios Core	26
3.3.3	Chemins importants dans l'environnement Nagios.....	27
3.4	Les compléments de Nagios.....	27
3.5	Implémentation de la solution	28
3.5.1	Ajout d'un utilisateur Nagios.....	29
3.5.2	Configuration des hôtes.....	29
3.5.3	Configuration des Services	31
3.5.4	Configuration de l'outil Thruk	32
3.5.5	Configuration de l'outil NagiosQL.....	33
3.6	Test de sécurité.....	34
3.6.1	Attaque Brute Force SSH avec Hydra	34
3.6.2	Attaque par déni de service (DoS) avec hping3	36
	Conclusion	37
4	Conception et développement de la plateforme de supervision	38
	Introduction.....	38

4.1	Les outils et technologies utilisés	38
4.2	Diagrammes UML	40
4.2.1	Diagramme de cas d'utilisation.....	40
4.2.2	Diagramme de séquence.....	42
4.3	Design de l'interface.....	44
4.3.1	Organisation générale de l'interface.....	44
4.3.2	Design des pages.....	44
4.3.3	Dynamisme et échanges avec le backend.....	51
4.3.4	Limites actuelles et pistes d'amélioration	51
	Conclusion	52
	Conclusion Générale.....	53
	Bibliographie	54
	Annexes	55
	Annexe 1 : Installation et Configuration de Nagios	56
	Annexe 2 : Installation et Configuration de NCPA	59
	Annexe 3 : Installation et Configuration de NagiosQL.....	63
	Annexe 4 : Installation et Configuration de Thruk.....	70

Liste des figures

Figure 1.1: Diagramme de GANTT	5
Figure 2.1: Vue globale d'un système de supervision	11
Figure 2.2: Supervision active.....	13
Figure 2.3: Supervision passive	13
Figure 2.4: Architecture de SNMP	14
Figure 2.5: Protocole SNMP : Les échanges entre le manager et l'agent SNMP.....	15
Figure 2.6: Tableau de bord de Zabbix	17
Figure 2.7: Page devices de Cacti	18
Figure 2.8: Tableau de bord de Icinga	18
Figure 2.9: Tableau de bord de Icinga	19
Figure 3.1: Architecture de Nagios	24
Figure 3.2: Architecture Globale de Supervision.....	29
Figure 3.4: Interface NCPA.....	30
Figure 3.5: Vue d'ensemble des hôtes supervisés dans Nagios	31
Figure 3.6: Vue d'ensemble des services supervisés dans Nagios	32
Figure 3.7: Interface web Thruk connectée à Nagios via Livestatus	33
Figure 3.8: Interface web NagiosQL	34
Figure 3.9: Commande nmap	35
Figure 3.10: Commande hydra	35
Figure 3.11: Contenu du auth.log.....	36
Figure 3.12: Commande hping3	36
Figure 4.1: Nagios	38
Figure 4.2: HTML	39
Figure 4.3: CSS	39
Figure 4.4: JavaScript.....	39
Figure 4.5: PHP	39
Figure 4.6: MYSQL	40
Figure 4.7: MK Livestatut.....	40
Figure 4.8: Diagramme de cas d'utilisation	41
Figure 4.9: Diagramme de séquence authentification	43
Figure 4.10: Diagramme de séquence consultation des services	43
Figure 4.11: Page authentification	44
Figure 4.12: Page d'accueil (tableau de bord)	45
Figure 4.13: En-tête d'état global	46
Figure 4.14: Fonctions de filtrage et de recherche.....	46

Figure 4.15: Tableaux hôtes.....	47
Figure 4.16: Tableaux services.....	47
Figure 4.17: Tableaux groupes hôtes.....	47
Figure 4.18: Tableaux groupes services.....	48
Figure 4.19: Page Problèmes	48
Figure 4.20: Page Historique	49
Figure 4.21: Page Profil.....	50

Liste des tableaux

Tableau 2.1: Types de réseaux.....	7
Tableau 2.2: Les médias de transmission	8
Tableau 2.3: Tableau de spécification de paramètre.....	20
Tableau 2.4: Paramètres de choix en fonction des solutions open source	20
Tableau 2.5: Somme des points pour chaque solution	21
Tableau 3.1: Ressources matérielles requises	26
Tableau 3.2: Chemins importants dans l'environnement Nagios.....	27
Tableau 3.3: Les compléments de Nagios	28

Liste des abréviations

Abréviation	Signification
SNMP	Simple Network Management Protocol
CPU	Central Processing Unit
RAM	Random Access Memory
API	Application Programming Interface
SSH	Secure Shell
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor (langage)
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
URL	Uniform Resource Locator
DNS	Domain Name System
JSON	JavaScript Object Notation
GUI	Graphical User Interface
NCPA	Nagios Cross Platform Agent
NagiosQL	Interface Web de gestion de Nagios
Thruk	Interface Web pour données Nagios

Introduction Générale

Dans un monde où les systèmes informatiques jouent un rôle central dans le fonctionnement des entreprises et des organisations, la supervision réseau devient un besoin crucial. La complexité croissante des infrastructures et la nécessité d'assurer une disponibilité continue des services exigent des solutions fiables, réactives et accessibles. C'est dans ce contexte que notre projet de fin d'études s'inscrit, avec pour objectif le déploiement d'une plateforme de supervision réseau basée sur l'outil open source Nagios.

Notre démarche s'articule autour de deux axes majeurs : d'une part, la mise en place technique d'un environnement de supervision complet, incluant l'installation, la configuration et l'intégration de modules complémentaires de Nagios ; d'autre part, le développement d'une interface web moderne et intuitive visant à simplifier la consultation des informations de supervision, même pour un public non technique.

Ce rapport présente de manière structurée l'ensemble des étapes de réalisation du projet, en commençant par les fondements théoriques de la supervision réseau, une étude comparative des outils existants, la justification du choix de Nagios, puis l'implémentation concrète de la solution jusqu'à la création d'une interface web personnalisée.

Dans ce présent rapport, nous allons aborder notre thème de manière structurée à travers quatre chapitres essentiels. Le premier chapitre est consacré au contexte général du projet, où nous présentons les motivations, les objectifs visés ainsi que la planification de notre travail. Le deuxième chapitre traite des concepts de base et de l'état de l'art, en mettant en lumière les notions fondamentales liées à la supervision des systèmes ainsi qu'une revue des solutions similaires afin de situer notre approche. Le troisième chapitre est dédié à l'implémentation de Nagios, où nous décrivons les étapes d'installation, de configuration, ainsi que les principaux plugins utilisés pour assurer une supervision efficace. Enfin, le quatrième chapitre porte sur la conception et le développement de la plateforme de supervision, en détaillant l'architecture retenue, les choix technologiques, les fonctionnalités développées et les résultats obtenus à travers des cas d'utilisation concrets.

1 Contexte général du projet

Introduction

Avec le développement rapide de l'informatique, les entreprises et les organisations utilisent de plus en plus de systèmes pour gérer leurs activités. Pour s'assurer que ces systèmes fonctionnent correctement, il est important de mettre en place une supervision. La supervision permet de surveiller en temps réel l'état des équipements et des services informatiques. Elle aide à détecter rapidement les problèmes et à garantir la sécurité, la performance et la disponibilité des ressources. Aujourd'hui, à cause de la complexité croissante des infrastructures informatiques, l'utilisation d'outils de supervision adaptés devient indispensable. Ce premier chapitre pose les bases du projet en présentant le contexte général, les motivations qui ont conduit à sa mise en œuvre ainsi que les objectifs poursuivis. Il aborde également la problématique à laquelle le projet tente de répondre, tout en exposant la planification des différentes étapes de son développement. Cette démarche vise à offrir une vision claire et structurée du cadre dans lequel s'inscrit le travail réalisé.

1.1 Présentation du projet

Le projet réalisé dans le cadre de ce travail consiste à mettre en place une solution de supervision réseau complète en s'appuyant sur l'outil open-source Nagios. Ce dernier, bien qu'étant l'un des systèmes de supervision les plus puissants et largement utilisés, souffre néanmoins d'une interface peu intuitive, souvent jugée complexe pour les utilisateurs non techniques. Ainsi, le projet s'est déroulé en deux grandes phases

Dans un premier temps, nous avons procédé à l'installation et la configuration de Nagios, en y intégrant plusieurs composants complémentaires pour enrichir ses fonctionnalités :

- ❖ **NagiosQL** pour simplifier la gestion des fichiers de configuration,
- ❖ **NCPA (Nagios Cross Platform Agent)** pour assurer une remontée précise des métriques système,
- ❖ **Thruk** comme interface d'analyse des données collectées.

Une fois le socle technique mis en place et fonctionnel, nous avons orienté la suite du projet vers un objectif essentiel : rendre la supervision accessible à tous. En effet, l'interface native de Nagios peut s'avérer difficile à utiliser pour un public non initié. Pour répondre à cette limite, nous avons conçu et développé une plateforme de visualisation personnalisée. Cette interface, moderne et intuitive, permet à tout utilisateur – qu'il soit technicien ou non – de consulter l'état

du système supervisé en temps réel, d'identifier facilement les anomalies, et de naviguer rapidement entre les différents services ou équipements.

Le projet vise ainsi à combiner la robustesse de Nagios avec l'ergonomie d'une interface moderne, afin de proposer une solution de supervision à la fois fiable, claire et accessible.

1.2 Problématique

Aujourd'hui, les entreprises et organisations s'appuient de plus en plus sur leurs réseaux informatiques pour assurer le bon fonctionnement de leurs services. Une interruption de service, une panne d'un serveur, ou un problème de communication entre les équipements peuvent avoir des conséquences importantes : perte de données, baisse de productivité, ou insatisfaction des utilisateurs.

Pour éviter cela, il est nécessaire de mettre en place une solution de supervision réseau qui permet de détecter rapidement les problèmes, d'envoyer des alertes aux responsables, et de garder une vue d'ensemble sur l'état du système informatique. Des outils comme Nagios sont très utilisés dans ce domaine, car ils sont puissants, gratuits et personnalisables. Cependant, leur utilisation peut être compliquée pour les personnes non techniques, car l'interface est souvent difficile à comprendre.

Dans notre projet, nous avons voulu répondre à une question simple mais essentielle :

Comment déployer une solution de supervision réseau avec Nagios qui soit à la fois efficace, personnalisable et facile à utiliser pour tout type d'utilisateur, même ceux qui ne sont pas experts en informatique ?

Cette problématique nous a amenées à chercher un bon équilibre entre la puissance technique de Nagios et la simplicité d'une interface moderne, capable de rendre la supervision accessible à tous.

1.3 Objectif du projet

Le principal but de ce projet est de concevoir et implémenter une solution de supervision de réseau qui permet de vérifier, en temps réel, l'état des composants informatiques, des services réseau, et des ressources système. Cela tente de satisfaire un besoin croissant en un outil de détection des pannes rapides, d'avertissement des responsables, et de présentation des conditions de santé du système d'information si possible.

Plus précisément, notre projet repose sur les objectifs suivants :

- **Déployer et configurer l'outil Nagios** dans un environnement de test, afin de surveiller différents hôtes et services essentiels tels que HTTP, SSH, PING, et SNMP.
- **Mettre en place des plugins personnalisés**, adaptés aux besoins spécifiques du système supervisé.
- **Assurer une gestion centralisée des alertes**, pour faciliter la détection et le traitement des incidents.
- **Développer une interface utilisateur de visualisation**, permettant une consultation claire, intuitive et structurée des informations issues de la supervision.
- **Garantir la fiabilité et la stabilité de la solution**, en veillant à une configuration optimale, à une surveillance continue et à une maintenance facilitée.

1.4 Planification

Cette section est dédiée à la présentation du processus de développement adopté ainsi qu'à la planification de notre projet

1.4.1 Planification du projet

La réussite d'un projet repose en grande partie sur une planification efficace du travail. Cela facilite l'établissement des différentes étapes de travail, l'attribution des tâches, la gestion des ressources et le suivi de progression de façon organisée. Dans cette partie, nous exposons la stratégie élaborée pour coordonner toutes les étapes de notre projet, depuis sa conception jusqu'à son aboutissement final.

1.4.2 Diagramme de Gantt

Le diagramme de GANTT est un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet. Il s'agit d'une représentation d'un graphe connexe, valué et orienté, qui permet de représenter graphiquement l'avancement du projet.

Le diagramme suivant représente le diagramme de GANTT qui montre la gestion et le déroulement de notre projet.

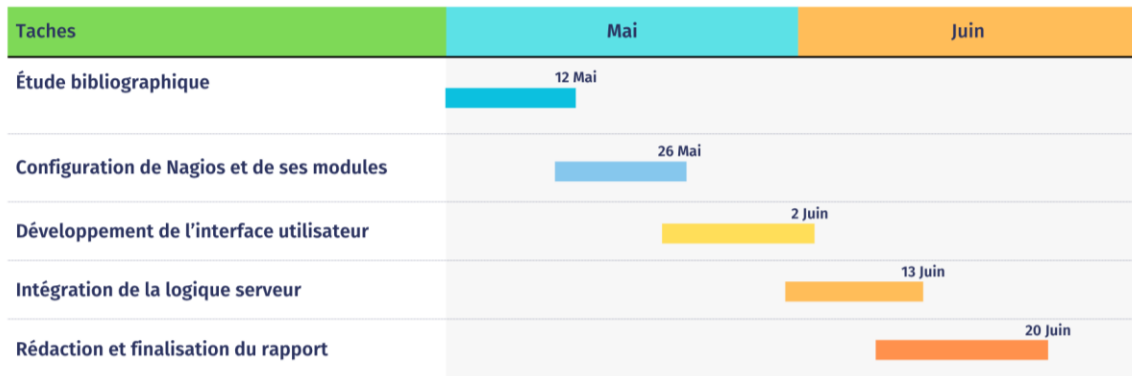


Figure 1.1: Diagramme de GANTT

Conclusion

Ce premier chapitre a permis de présenter le cadre général du projet, en exposant ses motivations, sa problématique, ses objectifs et sa planification. Face aux enjeux liés à la supervision des infrastructures informatiques, nous avons opté pour la mise en place d'une solution basée sur Nagios, enrichie par une interface moderne et accessible.

Le chapitre suivant abordera les concepts fondamentaux liés aux réseaux, à la sécurité et à la supervision, ainsi qu'une étude des outils existants, afin de mieux comprendre les choix techniques qui ont guidé notre implémentation.

2 Concepts de base et état de l'art

Introduction

Dans le cadre de ce projet, il est important de comprendre certains points essentiels, notamment ce qui touche aux réseaux informatiques, à la supervision, et aux différents outils utilisés dans ce domaine. On ne peut pas vraiment avancer dans la partie technique sans avoir un minimum de connaissances théoriques.

Ce chapitre servira donc à poser ces bases. On y parlera des éléments fondamentaux des réseaux, des aspects liés à la sécurité, du protocole SNMP qui est souvent utilisé pour la supervision, et aussi des outils les plus connus qu'on peut utiliser pour ce genre de projet.

2.1 Les réseaux informatiques

2.1.1 C'est quoi un réseau

Est un ensemble des appareils autonomes tels que des **ordinateurs**, des **imprimantes** et des **serveurs** connectés entre eux, et qui sont situés dans un certain domaine géographique. Permettant l'échange de données et le partage de ressources.

Par exemple, Dans un environnement connecté, les utilisateurs peuvent, accéder aux imprimantes partagées, utiliser des applications hébergées sur des serveurs distants, ou se connecter à Internet pour des communications internationales.

2.1.2 Nécessité d'un réseau

Le réseau informatique est nécessaire pour :

- Permettre la communication et l'échange de données entre plusieurs appareils
- La sécurisation des données
- Permettre l'administration à distance : accéder à des machines ou serveurs sans être physiquement présent.
- Faciliter la sauvegarde des données
- Superviser et gérer le système informatique : surveiller les équipements, détecter les pannes ou intrusions.

2.1.3 Types de réseaux

Les réseaux sont divisés selon leur passage géographique et les capacités des machines qui peuvent entraîner. Les trois types les plus courants sont :

- **LAN (Local Area Network):** constitué d'ordinateurs et de périphériques reliés entre eux et implantés dans une même entreprise, et à caractère privé. Il ne dépasse pas généralement la centaine de machines.
- **MAN (Metropolitan Area Network)** correspond à la réunion de plusieurs réseaux locaux (LAN) à l'intérieur d'un même périmètre d'une très grande Entreprise ou d'une ville
- **WAN (Wide Area Network):** un réseau multi-services couvrant un pays ou un groupe de pays, peut être privé ou publique.

Voici un tableau plus explicatif sur la différence entre chaque type :

Type de réseau	Distance approximative	Débit	Technologie utilisée	Exemple
LAN (Local Area Network)	Jusqu'à 1 km	Élevé (100 Mbps à 10 Gbps)	Ethernet, Wi-Fi	Réseau d'entreprise ou domestique
MAN (Metropolitan Area Network)	Jusqu'à 100 km	100 Mbps à plusieurs Gbps	Fibre optique, technologies MPLS	Réseau d'une ville ou d'un campus
WAN (Wide Area Network)	> 100 km	Variable (de 1 Mbps à plusieurs Gbps)	Fibre, satellite, lignes louées	Internet, réseau entre plusieurs sites d'entreprise

Tableau 2.1: Types de réseaux

2.1.4 Equipements Réseaux

Un réseau informatique est établi à l'aide des équipements réseaux qui sont les supports de gestion et de transmission des données entre les différents terminaux.

Ces équipements se divisent généralement en deux grandes catégories :

Les périphériques réseau, qui assurent la connectivité et la communication entre les appareils, et les médias de transmission, qui servent de support physique pour le transport des données.

➤ Les médias de transmission

Ce sont des canaux physiques nécessaires pour relier les différentes unités de communication. Ils sont caractérisés par leurs impédances caractéristiques et leurs bandes passantes.

Voici les principaux médias de transmission pertinents :




Image	Câble	description
	Coaxial	Il est composé de deux conducteurs cylindriques de même axe, l'âme et la tresse, séparés par un isolant. Ce dernier permet de limiter les perturbations dues aux bruits externes. Il a une Meilleure protection contre les interférences
	Fibre optique	Elle permet des vitesses de transmissions extrêmement rapides (jusqu'à 155 Mbps) très utiles pour les transferts d'images, vidéos, et pour le multimédia en général. Elle est constituée d'un fil de verre extrêmement fin.
	Paire torsadée (UTP/STP)	On l'utilise dans les réseaux locaux pour connecter les équipements tels que les serveurs, les switches et les routeurs. Ce câble est facile à installer.

Tableau 2.2: Les médias de transmission

➤ Les périphériques réseau

Ce sont les éléments matériels qui permettent la communication, la transmission et la gestion des données entre différents appareils. Ces équipements jouent un rôle clé dans la connectivité, la performance et la sécurité du réseau, et leur supervision est essentielle pour garantir un fonctionnement optimal.

- **Switch** ; connecte plusieurs appareils au sein d'un réseau local. Il gère des informations de routage limitées sur les nœuds du réseau interne et permet des connexions à des systèmes tels que les concentrateurs ou les routeurs
- **Routeur** : Il permet de faire circuler les paquets de données entre différents réseaux, notamment entre un réseau local (LAN) et Internet. Il analyse l'adresse de destination de chaque paquet et choisit le meilleur chemin pour l'acheminer.
- **Hub** : il permet de relier plusieurs appareils (ordinateurs, imprimantes, ...) dans un réseau local. Lorsqu'il reçoit un signal depuis un appareil connecté, il le retransmet à tous les autres ports, sans distinction. (Tous les appareils reçoivent les mêmes données.)
- **Serveur** : Ordinateur centralisé qui fournit des services, des ressources ou des données à d'autres

2.2 Généralités sur la sécurité réseau

La sécurité réseau regroupe l'ensemble des mécanismes et des bonnes pratiques visant à protéger les données et les ressources du réseau contre les accès non autorisés, les intrusions, les modifications ou les pertes.

2.2.1 Objectifs de la sécurité réseau

La sécurité réseau repose sur trois objectifs essentiels, souvent appelés **triade CIA** (Confidentiality, Integrity, Availability) :

La confidentialité : implique les efforts d'une organisation pour s'assurer que les données sont gardées secrètes ou privées. Pour ce faire, l'accès aux informations doit être contrôlé afin d'empêcher le partage non autorisé des données, qu'il soit intentionnel ou accidentel.

L'intégrité : a pour objectif d'assurer que les données ne soient ni altérées, ni supprimées, ni corrompues, volontairement ou accidentellement, au cours de leur transit ou de leur stockage.

La disponibilité : Même si les données sont tenues confidentielles et que leur intégrité est préservée, elles sont souvent inutiles à moins qu'elles ne soient disponibles pour les membres de l'organisation et les clients qu'elles servent. Cela signifie que les systèmes, réseaux et applications doivent fonctionner comme ils le devraient et quand ils le devraient. De plus, les personnes ayant accès à des informations spécifiques doivent être en mesure de les utiliser quand elles en ont besoin, et l'accès aux données ne devrait pas prendre trop de temps.

2.2.2 Typologie des menaces réseau

On parle de menaces quand quelque chose peut mettre en danger la sécurité, la disponibilité ou le bon fonctionnement des systèmes et des données. Ces menaces peuvent venir de différentes sources et chercher à atteindre des buts variés. Elles utilisent aussi plusieurs méthodes pour attaquer. Il est donc important de bien comprendre ces menaces pour pouvoir mieux les détecter et les empêcher.

On distingue plusieurs grandes catégories de menaces qui peuvent affecter un réseau. Nous allons nous concentrer sur deux types spécifiques que nous avons choisis pour les tests de sécurité réalisés avec Nagios.

- **Attaque par Force Brute**

L'attaque par brute force est une méthode qui consiste à essayer toutes les combinaisons possibles de mots de passe ou de clés pour accéder à un système. L'attaquant utilise généralement des outils automatisés qui testent rapidement un grand nombre de possibilités jusqu'à ce que le bon mot de passe soit trouvé. Ce type d'attaque cible souvent les systèmes d'authentification comme SSH ou les services web protégés par mot de passe. La réussite de cette attaque dépend largement de la complexité du mot de passe et des mécanismes de sécurité mis en place, comme la limitation du nombre de tentatives ou la mise en place d'un délai entre chaque essai.

- **Attaque par déni de service (DoS)**

L'attaque par déni de service (DoS) consiste à envoyer un grand nombre de requêtes ou de données vers un serveur ou un service pour le saturer. Le but est de l'empêcher de fonctionner normalement et de rendre le service inaccessible aux utilisateurs légitimes.

Quand trop de demandes arrivent en même temps, le serveur n'arrive plus à les traiter toutes, ce qui provoque un ralentissement ou un arrêt complet du service.

Parfois, cette attaque est lancée depuis un seul ordinateur, mais souvent elle est faite en même temps depuis plusieurs machines, ce qui complique sa détection et sa défense.

2.2.3 Sécurisation d'un réseau informatique

La sécurité d'un réseau informatique repose sur l'ensemble des mesures mises en place pour empêcher les accès non autorisés, protéger les données échangées, et garantir le bon fonctionnement des systèmes. Pour atteindre ces objectifs, plusieurs stratégies peuvent être mises en œuvre :

- **Contrôle des accès au réseau** : Il est fondamental de restreindre l'accès aux ressources uniquement aux utilisateurs et machines autorisés. Cela se traduit notamment par la mise en place de pare-feux (firewalls), de listes de contrôle d'accès (ACL) et de filtrages basés sur les adresses IP ou les ports.
- **Renforcement des mécanismes d'authentification** : L'usage de mots de passe complexes, renouvelés régulièrement, ainsi que le déploiement d'authentification multifacteur (MFA) sont recommandés pour sécuriser les points d'entrée sensibles tels que les connexions SSH ou les interfaces web d'administration.
- **Protection contre les attaques par déni de service (DoS)** : Des mécanismes de limitation du trafic (rate limiting), de détection d'anomalies réseau et l'usage de proxys ou de services de filtrage permettent d'atténuer l'impact des attaques visant à rendre un service indisponible.

2.3 La supervision réseau

2.3.1 Définition de la supervision réseau

La supervision réseau est l'ensemble des processus et technologies permettant la surveillance du bon fonctionnement de votre infrastructure de communication. Cette pratique essentielle consiste à monitorer en temps réel tous les composants de votre réseau informatique (routeurs, switches, serveurs, pare-feu) pour détecter rapidement les anomalies, pannes ou ralentissements. Un outil de supervision réseau collecte et analyse en continu les données de performance, génère des alertes automatiques en cas d'incident et fournit des tableaux de bord permettant une vision globale de l'état de santé de votre système.

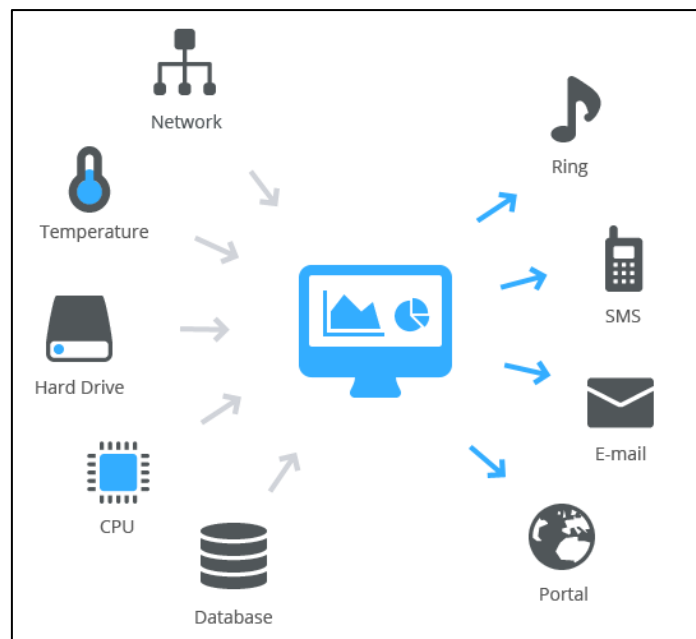


Figure 2.1: Vue globale d'un système de supervision

2.3.2 Principe

La supervision réseau peut être mise en œuvre sur la base d'analyse de résultats de commandes et de scripts locaux mais c'est surtout sur la base de protocoles standards. De nombreux logiciels existent et la communauté open source est particulièrement active dans le monitoring. Les logiciels permettent d'assister le technicien grâce à des interfaces de visualisations l'ensemble du réseau et à des alertes.

Des solutions logicielles proposant la supervision d'un réseau sont capables de vérifier l'état des équipements et des services à des intervalles de temps réguliers. Les données de résultats sont exploitables sous 3 formes différentes :

- Booléen (Le service est-il disponible ou non ?)
- Numérique (Quel est le temps de réponse de la machine ?)
- Qualitatif (Quel type d'erreur est renvoyé ?)

Les solutions de supervision permettent également de remplir des rapports d'activité selon la nature du service surveillé, comme des graphes d'utilisation réseau, ou encore des historiques de changement d'état sur le temps.

2.3.3 Objectifs de la supervision réseau

La supervision réseau a pour vocation de répondre à trois objectifs stratégiques pour toute organisation connectée :

✓ **Assurer la disponibilité des ressources**

Surveiller en temps réel les équipements et les flux réseau permet de minimiser les interruptions de service et d'assurer une connectivité continue, 24h/24.

✓ **Renforcer la sécurité du réseau**

En détectant des comportements anormaux (trafic inhabituel, équipements hors ligne, ports ouverts), la supervision contribue à identifier rapidement des menaces potentielles, comme des attaques DDoS ou des intrusions.

✓ **Optimiser les performances**

Une supervision efficace permet de mesurer et analyser les performances réseau (latence, bande passante, goulots d'étranglement...) et d'améliorer la qualité de service, notamment sur les applications critiques.

En résumé : la supervision réseau permet d'anticiper les incidents, de réduire les temps d'intervention, et d'améliorer durablement l'efficacité opérationnelle.

2.3.4 Les utilisateurs de la supervision réseau

- **Administrateurs réseau et administrateurs systèmes** : Les administrateurs réseau utilisent la supervision pour assurer la disponibilité, résoudre les incidents, et anticiper les dégradations sur les infrastructures techniques.
- **Ingénieurs DevOps / SRE** : Ils l'intègrent dans des pipelines d'automatisation, pour monitorer l'infrastructure as code, les conteneurs ou les environnements cloud.
- **Responsables IT / DSI** : Ils s'appuient sur les tableaux de bord stratégiques, les indicateurs de performance (KPI) et les rapports SLA pour piloter la qualité de service IT.
- **Équipes support et exploitation informatique** : Elles reçoivent les alertes en cas de problème et peuvent intervenir proactivement, réduisant ainsi les sollicitations utilisateur.
- **Prestataires IT et MSP (Managed Services Providers)** : Ils utilisent la supervision pour gérer à distance l'infrastructure de leurs clients et honorer leurs engagements contractuels (SLA).

2.3.5 Méthodes de supervision

Deux grandes méthodes de supervision sont utilisées avec plusieurs variantes : les méthodes active et passive, détaillées dans les paragraphes suivants :

Supervision active : La supervision active est la plus classique et la plus utilisée. Elle consiste en l'envoi de requêtes d'interrogation et de mesure par la plateforme de supervision. Elle a l'avantage d'être fiable : les vérifications se font de manière régulière et en mode question-réponse. Cette méthode est composée de trois étapes :

- Le serveur envoie une requête vers la ressource supervisée.
- La ressource répond à la requête du serveur.
- Le serveur analyse l'information et détermine un état pour la ressource.

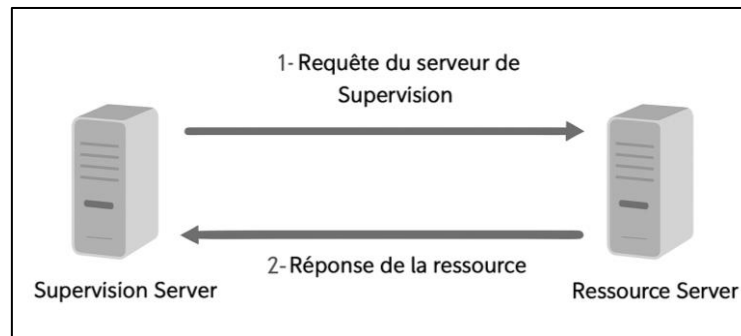


Figure 2.2: Supervision active

Le protocole le plus utilisé par les outils de supervision, SNMP utilise la méthode active.

Supervision passive : La supervision passive l'est du point de vue du serveur de supervision : ce sont les ressources supervisées qui transmettent des alertes au serveur de supervision :

- La ressource supervisée vérifie son état et transmet de manière autonome le résultat au serveur de supervision.
- Le serveur de supervision reçoit l'alerte et la traite.

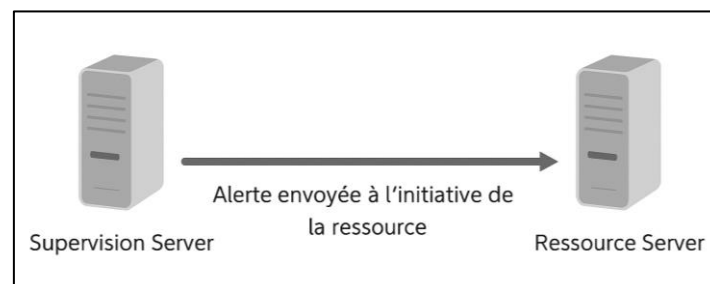


Figure 2.3: Supervision passive

Le protocole standardisé et privilégié pour la supervision passive est aussi SNMP avec le mécanisme de trappes.

2.4 Protocole SNMP

Pour bien superviser, les systèmes de supervision utilisent des protocoles. Nous allons étudier quelques protocoles de supervision

2.4.1 Définition du protocole SNMP

Le protocole SNMP est utilisé par la grande majorité des solutions de supervision. C'est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes de services du réseau, mais aussi de superviser un système d'exploitation, etc.

Pour cela, deux principes sont utilisés afin de récolter des informations :

- Requête du serveur vers l'équipement : supervision active.
- Alertes envoyées spontanément de l'équipement vers le serveur (traps) : supervision passive.

2.4.2 Les différentes versions du SNMP

Il existe actuellement 3 versions différentes du protocole SNMP :

- **SNMP v1** : première version standard mais très pauvre au niveau de la sécurité.
- **SNMP v2** : avec une amélioration de la sécurité mais jamais unifiée.
- **SNMP v3** : de nouveau standard avec une grosse évolution au niveau de la sécurité avec 2 concepts, USM pour utiliser le système nom d'utilisateur et un mot de passe cryptés en, et VACM pour une restriction de lecture de la MIB.

2.4.3 Architecture et composants

L'environnement de gestion SNMP est constitué de plusieurs composantes : La station de supervision (Manager), les éléments actifs du réseau, les variables MIB et des agents SNMP.

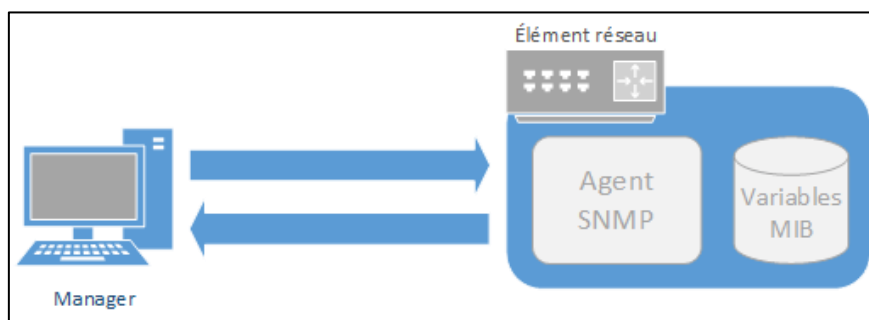


Figure 2.4: Architecture de SNMP

Les différentes composantes du protocole SNMP sont les suivantes :

- **Manager** : Il exécute les applications de gestion qui contrôlent les éléments réseaux. Physiquement, la station est un poste de travail. Le manager va aller récupérer les informations auprès des agents et les centraliser.

- **Elément du réseau** : Ce sont les équipements (Ex : Routeur, Switch, Poste de travail, imprimante, ...) que l'on cherche à gérer. Chaque élément réseau est composé d'un Agent SNMP et d'une variable MIB.
- **Agent SNMP** : Chaque élément du réseau dispose d'un agent SNMP qui répond aux requêtes du manager. Ils vont chercher l'information requise dans la MIB et la retransmettre ensuite au manager.
- **MIB**: C'est une collection d'objets représentant les caractéristiques du terminal administré.

2.4.4 Les requêtes SNMP

Le mécanisme de base du protocole SNMP est constitué d'échanges de type requête/réponse appelé PDU. Il existe quatre types de requêtes SNMP : GetRequest, GetNextRequest, GetBulk, SetRequest.

- ❖ La requête GetRequest qui recherche une variable sur un agent ;
- ❖ La requête GetNextRequest qui recherche la variable suivante ;
- ❖ La requête GetBulk qui recherche un ensemble de variables regroupées ;
- ❖ La requête SetRequest qui change la valeur d'une variable sur un agent.

Les réponses de SNMP :

À la suite de requêtes, l'agent répond toujours par GetResponse. Toutefois si la variable demandée n'est pas disponible, le GetResponse sera accompagné d'une erreur noSuchObject.

Les alertes (Traps) :

Les alertes sont envoyées quand un événement non attendu se produit sur l'agent. Celui-ci en informe la station de supervision via une trap. Donc une notification est envoyée vers son manager pour signaler un événement, un changement d'état ou un défaut. L'agent n'attend pas d'acquittement de la part du manager.

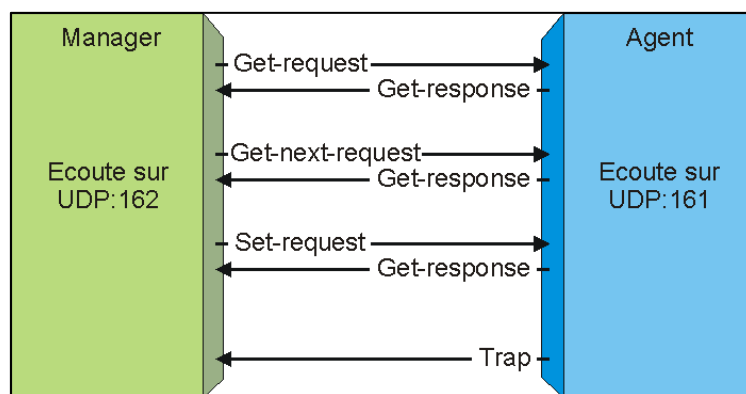


Figure 2.5: Protocole SNMP : Les échanges entre le manager et l'agent SNMP

2.4.5 Avantages et limites de SNMP

Le protocole SNMP est un outil largement utilisé pour la supervision des équipements réseau. Il permet de surveiller l'état des périphériques, de collecter des informations, et d'envoyer des alertes en cas de problème.

Avantages de SNMP :

- Compatible avec la majorité des équipements réseau (switches, routeurs, serveurs...)
- Facile à configurer et léger à déployer
- Permet la centralisation des données de supervision
- Supporte l'envoi d'alertes automatiques (traps)
- Basé sur un standard universel avec des MIBs pour chaque appareil

Inconvénients de SNMP :

- Sécurité faible dans les versions 1 et 2c (données non chiffrées)
- Repose sur UDP, donc perte possible de paquets
- Difficulté à comprendre ou personnaliser certaines MIBs
- Moins adapté à la configuration ou au contrôle distant
- Moins performant dans les très grands réseaux mal optimisés

2.5 Les outils de supervision existants

2.5.1 Introduction aux outils de supervision réseau

Les outils de supervision réseau jouent un rôle essentiel dans la gestion et le maintien en conditions opérationnelles des infrastructures informatiques. Ils permettent de surveiller en temps réel l'état des équipements (serveurs, routeurs, switches, etc.), de détecter les anomalies, de générer des alertes en cas d'incident, et d'analyser les performances du système.

Grâce à ces outils, les équipes IT peuvent anticiper les défaillances, réagir rapidement en cas de problème, et garantir un haut niveau de disponibilité des services. La supervision contribue ainsi à renforcer la sécurité, à améliorer la qualité de service et à optimiser l'exploitation des ressources.

Il existe de nombreuses solutions de supervision, allant des outils simples spécialisés dans les graphes de performances, jusqu'aux plateformes complètes capables de superviser des environnements complexes et hybrides (physique, virtuel, cloud). Parmi les outils les plus utilisés, on retrouve Zabbix, Cacti, Icinga, Centreon, ou encore Nagios.

2.5.2 Exemples d'outils populaires

Plusieurs outils de supervision réseau sont aujourd'hui largement utilisés en entreprise. Chacun possède ses spécificités en termes de fonctionnalités, d'interface, de protocoles supportés et de cas d'usage. Voici une présentation détaillée des quatre outils les plus connus :

a. Zabbix

Zabbix est une solution de supervision open source particulièrement complète, capable de surveiller en temps réel les performances des infrastructures réseau, des serveurs, des bases de données, des applications ou encore des machines virtuelles. Il utilise divers protocoles (SNMP, IPMI, JMX, agents natifs, scripts personnalisés) pour collecter les données et fournir des tableaux de bord dynamiques.

Son principal atout est son interface web moderne et intuitive, permettant de créer facilement des vues personnalisées, des cartes réseau, des alertes conditionnelles et des rapports. Zabbix se distingue également par ses capacités avancées en matière de détection automatique des hôtes, de corrélation d'événements et d'escalade des alertes.

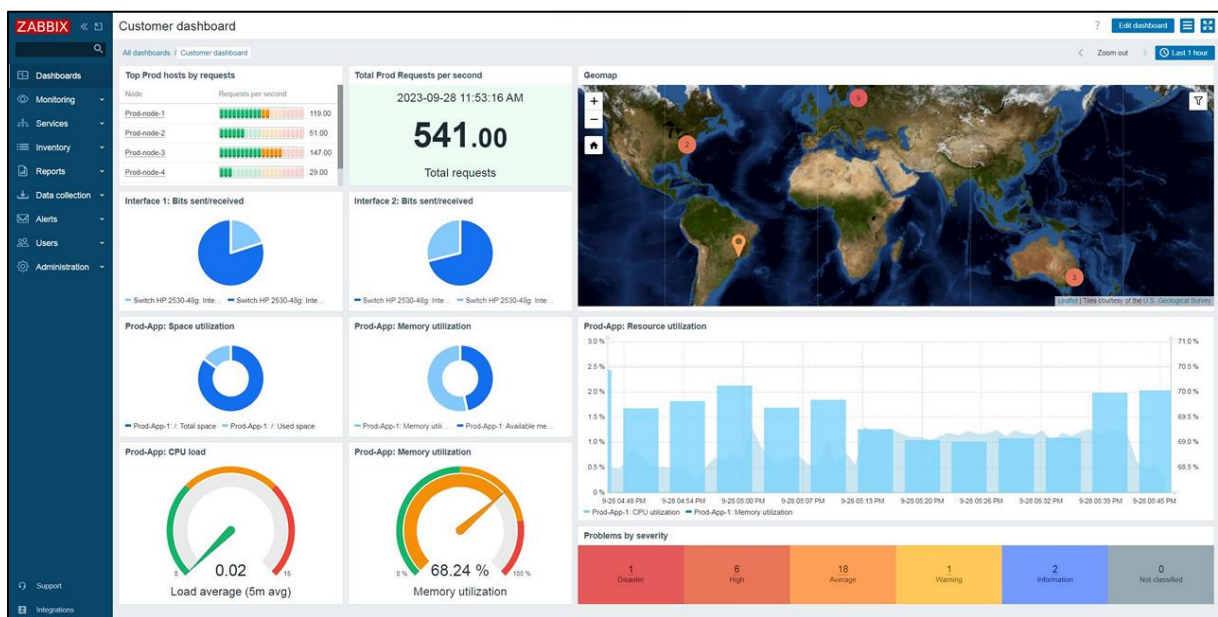


Figure 2.6: Tableau de bord de Zabbix

b. Cacti

Cacti est un outil spécialisé dans la supervision graphique, principalement basé sur le protocole SNMP. Il repose sur RRDTool pour stocker les données de performance et générer des graphes détaillés dans le temps (trafic réseau, charge CPU, utilisation mémoire, etc.).

Bien qu'il n'intègre pas de moteur de supervision aussi puissant que d'autres outils comme Nagios ou Zabbix, Cacti est très apprécié pour sa simplicité de configuration et sa capacité à offrir une vision claire de l'évolution des métriques. Il est particulièrement adapté aux administrateurs souhaitant une visualisation précise et rapide de l'état de leurs équipements réseau.

Device Description	Hostname	ID	Graphs	Data Sources	Status	In State	Uptime	Poll Time	Current (ms)	Average (ms)	Availability	Created
Cacti Server	localhost	1	4	5	Up	N/A	N/A	0.1	0	0	100 %	2020-09-06 21:43:06
Central NAS	192.168.11.105	56	12	19	Up	120	42	0.26	0.35	1.15	99.36 %	2020-09-06 21:43:06
HP Printer	192.168.11.174	55	22	22	Up	137	54	0.65	1.04	1.8	99.61 %	2020-09-06 21:43:06
vhost01	192.168.11.201	46	12	19	Up	120	4	0.38	1.45	1.61	99.99 %	2020-09-06 21:43:06
vhost02	192.168.11.202	45	12	19	Up	120	4	0.34	0.56	0.94	99.99 %	2020-09-06 21:43:06
vhost03	192.168.11.203	44	12	19	Up	120	4	0.24	0.9	2.09	99.98 %	2020-09-06 21:43:06
vhost04	192.168.11.204	43	12	19	Up	120	4	0.26	1.01	0.76	100 %	2020-09-06 21:43:06
vhost05	192.168.11.205	42	12	19	Up	120	4	0.33	0.83	1.25	99.99 %	2020-09-06 21:43:06
vhost06	192.168.11.206	41	12	19	Up	120	4	0.39	0.74	0.79	100 %	2020-09-06 21:43:06
vhost07	192.168.11.207	40	12	19	Up	267	4	0.4	0.52	1.06	98.93 %	2020-09-06 21:43:06
vhost08	192.168.11.208	39	12	19	Up	120	4	0.19	0.89	1.24	99.99 %	2020-09-06 21:43:06
vhost09	192.168.11.209	38	12	19	Up	267	4	0.15	0.7	1.07	98.93 %	2020-09-06 21:43:06
vhost10	192.168.11.210	37	12	19	Up	120	4	0.22	0.77	0.77	100 %	2020-09-06 21:43:06
vhost11	192.168.11.211	36	12	19	Up	120	4	0.09	2.61	1.01	99.98 %	2020-09-06 21:43:06
vhost12	192.168.11.212	35	12	19	Up	120	4	0.32	1.14	1.09	99.99 %	2020-09-06 21:43:06
vhost13	192.168.11.213	34	12	19	Up	120	4	0.25	2.63	1.05	99.98 %	2020-09-06 21:43:06
vhost14	192.168.11.214	33	12	19	Up	267	4	0.26	3.99	1.02	98.93 %	2020-09-06 21:43:06
vhost15	192.168.11.215	32	12	19	Up	120	4	0.31	1.11	0.93	99.99 %	2020-09-06 21:43:06

Figure 2.7: Page devices de Cacti

c. Icinga

Icinga est né comme un fork de Nagios, avec l'objectif de moderniser l'architecture et l'interface utilisateur. Il est conçu pour surveiller les hôtes et services, générer des alertes, et produire des rapports d'état détaillés. Icinga se décline aujourd'hui en deux versions : Icinga 1 (proche de Nagios) et Icinga 2, beaucoup plus avancée, modulaire et orientée vers les environnements complexes.

Grâce à son API REST, Icinga permet une intégration facile avec d'autres outils comme Grafana (pour la visualisation), Elasticsearch ou Prometheus. Il est apprécié pour son architecture flexible, sa gestion fine des alertes et son interface web responsive.

Service Problems	Host Problems	Cube
CRITICAL SSH on Localhost connect to address 127.0.0.2 and port 22: Connection refused CRITICAL Swap on Localhost swap CRITICAL - 0% free (0 MB out of 8 MB) - Swap is either disabled, not present, or of zero size. CRITICAL DNS on DNS Server 2 60000: srv-dns2.icinga.com was not found by the server WARNING Disk on Localhost disk WARNING - free space: /etc/passwd.conf 20041 MB (20% free of 100 MB) - /etc/passwd.conf 20041 MB (20% free of 100 MB) CRITICAL MySQL on MySQL Development (DOWN) CRITICAL - connection could not be established within 60 seconds CRITICAL SSH on MySQL Server (DOWN) CRITICAL - Socket timeout after 10 seconds CRITICAL SSH on Webserver Testing (DOWN) CRITICAL - Socket timeout after 10 seconds CRITICAL SSH on Icinga Exchange (DOWN) CRITICAL - Socket timeout after 10 seconds CRITICAL SSH on MySQL Development (DOWN) CRITICAL - Socket timeout after 10 seconds CRITICAL HTTP on Icinga Exchange (DOWN) HTTP CRITICAL: HTTP/1.1 200 OK - string "welcome to Icinga Exchange" not found on /http://exchange.icinga.com/001 - 31376 bytes in 0.005 second response time	DOWN Icinga Exchange PING CRITICAL - Packet loss = 100% DOWN MySQL Development PING CRITICAL - Packet loss = 100% DOWN MySQL Server PING CRITICAL - Packet loss = 100% DOWN Webserver Development PING CRITICAL - Packet loss = 100% DOWN Webserver Testing PING CRITICAL - Packet loss = 100%	development (2) ↑ test1 1 new-pork 1 production (3) ↑ test1 1 test2 1 new-pork 1 testing (2) ↑ test1 1

Figure 2.8: Tableau de bord de Icinga

d. Centreon

Centreon est une solution de supervision française, construite initialement autour du moteur de Nagios, mais qui a fortement évolué pour offrir une plateforme complète, professionnelle et accessible. Elle propose une interface web ergonomique, un système de configuration simplifié, et une large gamme de plugins prêts à l'emploi.

Centreon est souvent utilisé dans les entreprises et collectivités pour sa facilité d'administration, ses capacités de reporting SLA/KPI, et son intégration avec les processus ITIL. Il peut superviser à la fois les équipements réseau, les serveurs, les bases de données et les applications critiques, et il dispose d'un support commercial pour les environnements les plus exigeants.

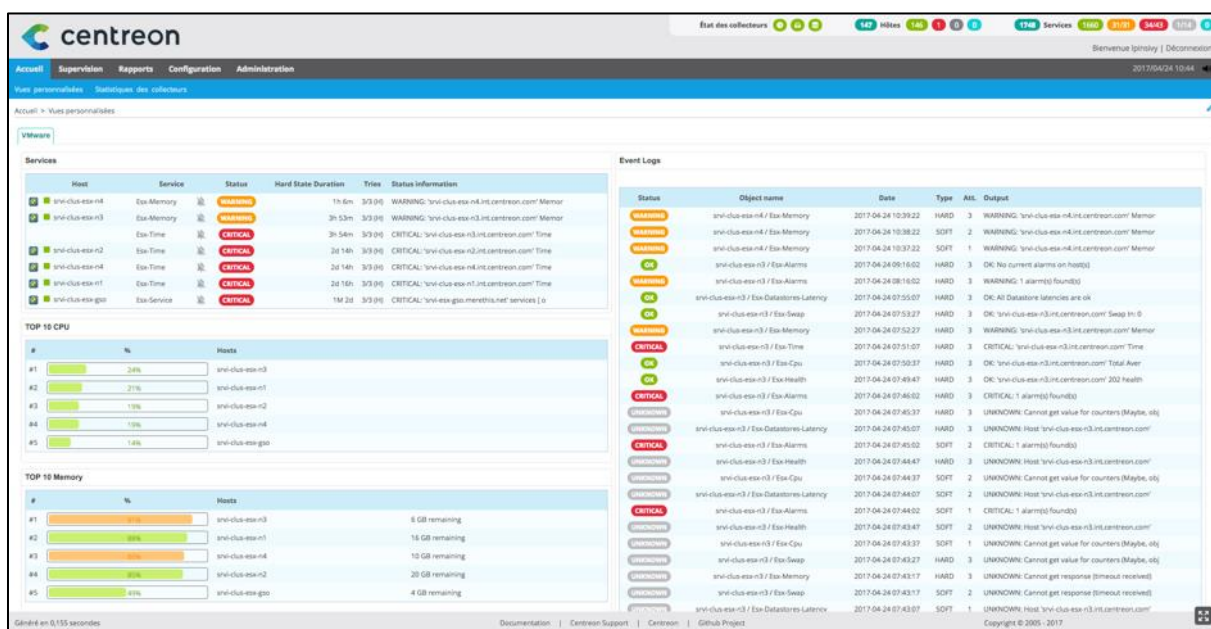


Figure 2.9: Tableau de bord de Icinga

2.5.3 Benchmarking des solutions

Dans le tableau suivant nous proposerons des paramètres de choix qui nous permettront de faire le bon choix d'une solution, entre les solutions citées précédemment :

Critère	1	2	3	4	5
Base de Nagios	Non	Oui			
Performance	Faible	Moyenne	Bonne	Très bonne	Excellent
Installation & configuration	Très difficile	Compliqué	Moyenne	Facile	Très simple
Utilisation des ressources	Peu	Moyenne	Bonne	Très bonne	Excellent

Dépendance entre services	Non	Oui			
Écriture et ajout de plugins	Non	Oui			
Interface web ergonomique	De base	Moderne			
Protocole SNMP	Non	Oui			
Communauté & documentation	Faible	Moyenne	Bonne	Très bonne	Excellente
Disponibilité des plugins	Faible	Moyenne	Large	Très large	Intégrée nativement
Simplicité d'extension personnalisée	Très difficile	Moyenne	Facile	Très facile	Native
Monitoring distribué	Non	Oui			

Tableau 2.3: Tableau de spécification de paramètre

Le tableau suivant illustre les paramètres de spécification en fonctions des outils de supervision open source, décrite précédemment :

Critère	Nagios	Zabbix	Centreon	Icinga	Cacti
Base de Nagios	2	1	2	2	1
Performance	3	3	4	4	2
Installation & configuration	4	3	4	2	3
Utilisation des ressources	2	3	2	3	3
Dépendance entre services	2	1	2	2	1
Ecriture et ajout de plugins	2	2	2	2	2
Interface web ergonomique	1	2	2	2	1
Protocole SNMP	2	2	2	2	2
Communauté & documentation	5	3	3	2	2
Disponibilité des plugins	4	3	3	2	1
Simplicité d'extension personnalisée	4	2	2	2	1
Monitoring distribué	2	1	2	2	2

Tableau 2.4: Paramètres de choix en fonction des solutions open source

En se basant, sur les spécifications des paramètres de choix, représentées dans le Tableau 1 et les paramètres de choix en fonction des solutions open source citées dans le Tableau 2, le calcul de la somme des points pour chaque solution, nous donnera le résultat représenté dans le Tableau 3 :

Nagios	Zabbix	Centreon	Icinga	Cacti
33	26	30	27	21

Tableau 2.5: Somme des points pour chaque solution

Le tri des solutions, en se basant sur le Tableau 3, nous a permis de classer les outils du plus adapté au moins pertinent selon les critères étudiés : Nagios, Centreon, Icinga, Zabbix et en dernier Cacti.

Nous écartons Cacti, car il ne propose qu'une supervision graphique basique sans moteur de surveillance avancé. Zabbix, bien que complet, demande une configuration plus complexe et dispose d'une interface encore perfectible selon nos besoins. Centreon et Icinga, bien qu'efficaces, présentent soit des dépendances fortes à d'autres outils (comme Nagios), soit une complexité d'installation plus importante dans le cas d'environnements simples.

Après cette étude, il nous semble que Nagios est la solution la plus adaptée à notre projet, en raison de sa robustesse, de sa flexibilité et de sa large communauté d'utilisateurs.

- ✓ Nagios est entièrement open source et largement utilisé dans l'industrie.
- ✓ Il est simple à mettre en œuvre, extensible par des plugins personnalisés.
- ✓ Il bénéficie d'une grande disponibilité de ressources, de plugins, et d'une documentation riche.
- ✓ Il offre une excellente compatibilité avec les protocoles standards comme SNMP.

Conclusion

Ce chapitre a permis de poser les fondations nécessaires à la compréhension du projet, en abordant les notions essentielles liées aux réseaux, à la sécurité, à la supervision, ainsi qu'aux outils les plus utilisés dans ce domaine. L'analyse comparative a mis en évidence Nagios comme la solution la plus adaptée à nos besoins, grâce à sa robustesse, sa compatibilité avec les protocoles standards comme SNMP, et sa grande flexibilité.

Le chapitre suivant sera consacré à l'implémentation de la solution Nagios, en détaillant les étapes de son installation, sa configuration, ainsi que son intégration dans notre infrastructure.

3 Implémentation de la solution Nagios

Introduction

Imaginez que vous êtes administrateur réseau dans une entreprise. Un matin, plusieurs utilisateurs vous signalent qu'ils n'arrivent pas à accéder à leurs e-mails. D'autres se plaignent d'une lenteur inhabituelle du réseau. Vous commencez à chercher l'origine du problème : est-ce un serveur qui ne répond plus ? Un service qui s'est arrêté ? Un routeur déconnecté ? Vous perdez un temps précieux à vérifier chaque équipement, sans avoir exactement par où commencer. Pendant ce temps, l'activité de l'entreprise est paralysée.

Maintenant, imaginez la même situation... mais cette fois, vous avez un système de supervision **Nagios**. Dès qu'un service tombe, vous recevez une alerte par e-mail. En quelques minutes, vous savez **exactement quel équipement ou service est en cause**, et vous pouvez intervenir immédiatement. Ce gain de temps, de réactivité et de visibilité est précisément ce que permet un outil de supervision bien configuré.

C'est dans cette optique que nous avons entrepris l'implémentation de **Nagios Core**. Ce chapitre présente la solution que nous avons choisie pour surveiller un réseau de manière centralisée, en assurant un suivi en temps réel de l'état des machines et des services. Nous allons explorer son architecture, son environnement, ses composants essentiels, ainsi que les étapes concrètes de son installation et de sa configuration dans le cadre de notre projet, tout en abordant les mécanismes permettant de sécuriser ce système face aux menaces potentielles.

3.1 Architecture de Nagios

Avant de passer à l'implémentation, il est important de comprendre comment les différents éléments de Nagios interagissent entre eux pour assurer une surveillance efficace. Cette section vous plonge au cœur de cette architecture, en expliquant le rôle de chaque composant et leur contribution à l'ensemble du système.

➤ Ordonnanceur (Nagios Core)

L'ordonnanceur est le cœur du système de supervision chargée de coordonner toutes les opérations de supervision. Son rôle est d'interpréter les fichiers de configuration dans lesquels sont définis les hôtes, les services, les contacts et les commandes. Plutôt que d'effectuer lui-même les contrôles, il délègue cette tâche à des plugins externes qui effectuent les vérifications et retournent un état précis (OK, Warning, Critical, Unknown).

➤ Les plugins

Nagios effectue tous ses contrôles à l'aide de plugins. Il s'agit de composants externes auxquels Nagios transmet des informations sur les éléments à vérifier, ainsi que sur les limites d'avertissement et de critique. Les plugins sont chargés d'effectuer les contrôles et d'analyser les résultats. Le résultat de ces contrôles est le statut (OK, Warning, Critical, Unknown) et un texte complémentaire décrivant en détail le service. Ce texte est principalement destiné aux administrateurs système pour leur permettre de consulter l'état détaillé d'un service.

Nagios est livré avec un ensemble de plugins standards qui permettent de vérifier les performances de presque tous les services que vous pourriez utiliser (nous verrons des exemples plus tard).

➤ L'interface Web

Il permet aux administrateurs et techniciens de visualiser l'état du réseau en temps réel. Accessible via un navigateur, cette interface affiche les hôtes et services surveillés, leurs états actuels, l'historique des alertes. Elle permet de désactiver temporairement certains contrôles ou notifications, et d'accéder aux détails de chaque alerte. Cette interface est généralement hébergée sur un serveur Apache et sécurisée par un système d'authentification.

➤ Le système de notification

Il permet d'alerter immédiatement les administrateurs ou les équipes techniques lorsqu'un problème est détecté sur un hôte ou un service supervisé. Les notifications sont configurables selon différents critères, tels que le type d'événement, les horaires de travail, ou encore les destinataires spécifiques. Nagios supporte plusieurs méthodes de notification, dont les emails, les SMS ou les scripts personnalisés.

➤ Les fichiers de configuration

C'est là qu'on va définir tout ce qu'on veut surveiller : les machines, les services, les groupes d'éléments, les personnes à prévenir, et toutes les commandes nécessaires. Ces fichiers sont écrits en texte simple, mais bien organisés, ce qui permet d'adapter Nagios exactement à ce dont on a besoin. Grâce à cette configuration, Nagios sait quand et quoi vérifier, qui prévenir en cas de problème, et comment réagir selon la situation. Pour bien utiliser Nagios, il faut vraiment comprendre comment créer et gérer ces fichiers, car c'est grâce à eux que la supervision fonctionne correctement.

➤ Base de données (optionnelle)

Nagios peut intégrer une base de données de type *MySQL* ou *PostgreSQL* pour y stocker des informations de supervision. Bien que conseillée, la base de données n'est pas essentielle dans le fonctionnement de Nagios et peut être remplacée par de simples fichiers tournants. Le choix de mettre en place une base de données dépend de l'utilisation qui sera faite de Nagios et des données collectées.

➤ Les agents distants

Quand on veut surveiller des machines à distance ou du matériel spécifique, Nagios utilise des agents comme NRPE ou des protocoles comme SNMP. NRPE, par exemple, permet à Nagios de lancer des contrôles directement sur la machine distante ça sert à vérifier des trucs comme l'utilisation du processeur, la mémoire, les processus en cours, ou encore les logs. SNMP, lui, est plutôt utilisé pour récupérer des infos sur les équipements réseau comme les routeurs, les switches ou les imprimantes. Ces agents font un peu le lien entre Nagios et les machines surveillées, ce qui permet de garder un œil sur tout, même quand ce n'est pas sur le serveur principal.

L'architecture standard de Nagios peut donc être représentée de la manière suivante :

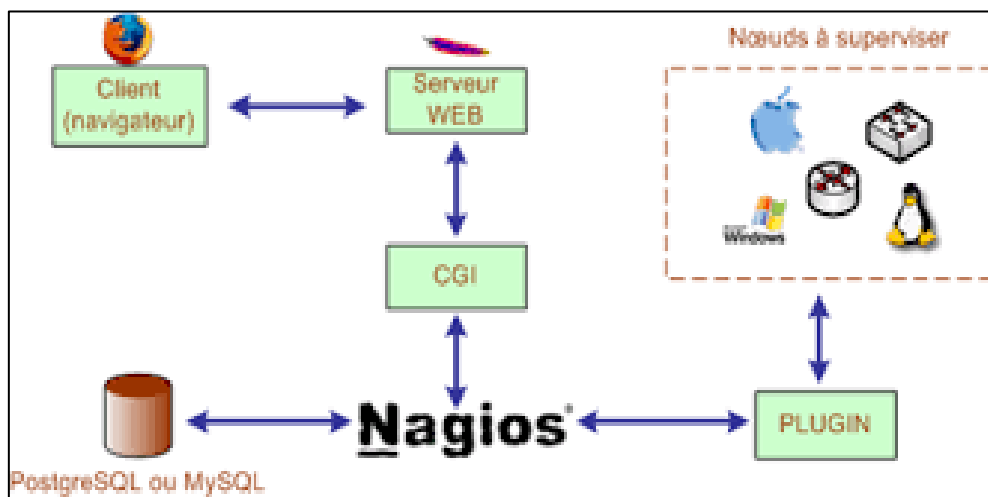


Figure 3.1: Architecture de Nagios

3.2 Présentation de Nagios

Nagios est un outil open source de surveillance système. Il surveille les serveurs et autres périphériques de votre réseau et s'assure de leur bon fonctionnement. Nagios vérifie en permanence le bon fonctionnement des autres machines et de leurs différents services. De plus, Nagios peut recevoir des informations d'autres processus ou machines concernant leur état ; par exemple, un serveur web peut lui envoyer des informations en cas de surcharge.

Son objectif principal est de détecter si un système ne fonctionne pas correctement dès que possible et d'en informer le personnel approprié, et si possible, d'essayer de résoudre l'erreur, par exemple en redémarrant les services système si nécessaire.

La surveillance du système dans Nagios est divisée en deux catégories d'objets, **les hôtes** et **les services** :

- Les hôtes représentent un périphérique physique ou virtuel sur votre réseau (serveurs, routeurs, postes de travail, imprimantes, etc.)
- Les services sont des fonctionnalités particulières, par exemple, un serveur SecureShell (SSH) peut être défini comme un service à surveiller

Chaque service est associé à un hôte sur lequel il s'exécute. De plus, les machines peuvent être regroupées en groupes d'hôtes.

3.2.1 Les avantages du Nagios

L'un des principaux avantages des contrôles de performance de Nagios est qu'ils n'utilisent que quatre états distincts : Ok, Warning, Critical, et Unknown.

Cette approche permet aux administrateurs d'ignorer les valeurs de surveillance et de se contenter de définir les limites Warning/Critical. Ce concept a fait ses preuves et est bien plus efficace que les graphiques de surveillance et l'analyse des trends. Il est similaire aux feux de signalisation : le vert indique qu'un service fonctionne correctement, l'état Warning correspond au feu jaune et l'état Critical au feu rouge.

En plus de cette simplicité est sa **structure basée sur des plugins**, vous permettant de développer votre propre plugin. Ainsi, si vous souhaitez vérifier quelque chose qui n'est pas encore réalisable, il suffit de développer un petit script personnalisé, puis de l'intégrer à l'environnement Nagios.

Un autre avantage est un rapport clair indiquant que les services X sont up et running, Y sont en état warning et Z sont actuellement critical, ce qui est beaucoup plus lisible qu'une matrice de valeurs. Cela vous fait gagner du temps dans l'analyse de ce qui fonctionne et de ce qui ne fonctionne pas. Cela peut également vous aider à prioriser les tâches à traiter en premier et les problèmes à traiter ultérieurement.

3.3 L'environnement de Nagios

Pour que l'installation du Nagios se passe bien, il est important de bien connaître les prérequis nécessaires à son bon fonctionnement. En effet, Nagios n'est pas un programme qu'on installe et qui fonctionne tout seul, il dépend d'un ensemble d'éléments qu'il faut préparer. Si tout cela n'est pas bien mis en place dès le départ, on risque d'avoir une supervision instable.

3.3.1 Prérequis matériels et logiciels

➤ Systèmes d'exploitation supportés

Nagios est initialement conçu pour fonctionner sous environnement Linux. Il s'installe généralement sur des distributions stables et répandues telles que Ubuntu, Debian, CentOS, ou encore RHEL. Cependant, il est également capable de superviser des machines Windows à travers l'utilisation d'agents dédiés.

Dans notre cas, nous avons choisi **Ubuntu** comme SE hôte pour plusieurs raisons :

- Ubuntu est une distribution largement documentée, ce qui facilite la recherche de solutions en cas de problèmes lors de l'installation ou de la configuration.
- Les versions LTS (Long Term Support) d'Ubuntu, comme la 22.04 ou la 24.04, offrent une stabilité et une sécurité accrues grâce à des mises à jour régulières sur plusieurs années.
- La gestion des paquets via apt rend l'installation des dépendances et des services nécessaires (comme Apache, PHP ou les outils de compilation) rapide.

➤ **Paquets requis pour l'installation :**

Nagios nécessite un certain nombre de dépendances pour fonctionner correctement. Parmi ces dépendances, nous avons choisi d'installer **PHP 8.4**.

Le serveur web **Apache2** est également essentiel. C'est lui qui permet d'héberger l'interface web de Nagios, grâce à laquelle on peut consulter les états des hôtes et services, visualiser les alertes en temps réel, et accéder à tous les rapports.

Pour installer Nagios à partir du code source, on a aussi besoin de **gcc** et **make**. Ces outils servent à compiler le logiciel et à assembler les différents modules nécessaires à son fonctionnement.

D'autres outils comme **perl**, **wget**, ou **tar** complètent cette liste en offrant des fonctionnalités d'extraction, de téléchargement et d'exécution de scripts, souvent nécessaires lors de l'installation de plugins ou de mises à jour.

➤ **Ressources matérielles requises**

Composant	Petite infrastructure (≤ 50 hôtes)	Moyenne infrastructure (≤ 200 hôtes)	Grande infrastructure (≥ 200 hôtes)
Processeur (CPU)	1 à 2 cœurs	2 à 4 cœurs	4 cœurs au plus
Mémoire vive (RAM)	2 à 4 Go	4 à 8 Go	8 à 16 Go
Stockage	10 à 20 Go	20 à 50 Go	50 Go ou plus

Tableau 3.1: Ressources matérielles requises

3.3.2 Installation de Nagios Core

Nagios lui-même, ainsi que les plugins standards, sont écrits en langage C et doivent être compilés en binaire natif pour fonctionner sous Linux. Pour cela, ils s'appuient sur la méthode classique « configure, make, make install », une procédure largement adoptée pour l'installation de nombreuses applications Unix développées en C. Les différentes étapes pratiques de cette installation sont détaillées dans [l'annexe 1 : Installation et configuration du Nagios]. Celle-ci

couvre notamment l'installation des paquets nécessaires, la compilation et l'installation de Nagios, la création d'un utilisateur dédié avec mot de passe, l'ajout et la configuration des plugins officiels, ainsi que l'accès final à l'interface web permettant de superviser les services.

3.3.3 Chemins importants dans l'environnement Nagios

Maintenant que Nagios est installé, avant de passer à son implémentation, il est essentiel de bien comprendre certains chemins clés créés durant l'installation. Ces chemins sont indispensables non seulement pour la configuration de Nagios, mais aussi pour l'intégration d'outils complémentaires que nous verrons plus en détail dans la section [3.4 Les compléments de Nagios](#). Les connaître permet d'éviter les erreurs, de naviguer facilement dans l'environnement, et de gagner du temps lors des ajustements ou du débogage.

Path	Description
/usr/local/nagios	Répertoire racine de Nagios. Il contient l'ensemble de l'environnement Nagios,
/usr/local/nagios/etc	Dossier de configuration principale de Nagios. On y trouve notamment : nagios.cfg : fichier de configuration global, objects/ : définitions des hôtes, services, commandes, contacts, etc.
/usr/local/nagios/sbin/	Contient les fichiers CGI utilisés pour l'interface web de Nagios.
/usr/local/nagios/var/	Répertoire contenant les fichiers temporaires et d'état, comme les logs, le fichier status.dat, les fichiers de performances, etc.

Tableau 3.2: Chemins importants dans l'environnement Nagios

3.4 Les compléments de Nagios

En plus de l'interface web par défaut, Nagios peut être géré à l'aide d'autres interfaces web plus avancées, qui facilitent la configuration, la visualisation des performances et l'administration du système de supervision. Plusieurs outils complémentaires ont été développés dans ce but, chacun répondant à un besoin spécifique.

Le tableau ci-dessous présente quelques-uns des outils les plus connus dans l'écosystème Nagios :

Type d'outil	Outil	Fonction principale
Dashboards	Thruk	Interface web plus simple que celle de base.
	NagVis	Affiche l'état du réseau sous forme de cartes graphiques.
	Centreon	Plateforme complète pour gérer, visualiser et suivre les alertes plus facilement.
	Nagiosgraph	Montre l'évolution des performances avec des courbes simples.
Agents de collecte	NSClient++	Utilisé sur Windows pour envoyer les infos système à Nagios.
	NRPE	Exécute des commandes sur les machines Linux pour récupérer des données.
	NCPA	Agent léger pour surveiller les ressources (CPU, RAM, disque...) sur tous les systèmes.
Configuration	NagiosQL	Permet de configurer Nagios via une interface web, sans modifier les fichiers à la main.

Tableau 3.3: Les compléments de Nagios

Dans notre projet, nous avons choisi d'intégrer trois outils complémentaires à Nagios Core pour améliorer sa gestion et sa visualisation.

- **NagiosQL** : pour faciliter la configuration de Nagios. Cet outil nous permet de créer et modifier les fichiers de configuration via une interface web, ce qui évite les erreurs manuelles.
- **Thruk** : parce qu'il offre une interface plus moderne et plus agréable que celle par défaut. Il rend la navigation et le suivi des services beaucoup plus simples.
- **NCPA** : nous l'avons utilisé comme agent de supervision, car il est facile à installer et compatible avec plusieurs systèmes. Il permet de surveiller les ressources.

3.5 Implémentation de la solution

Maintenant que notre système Nagios est correctement installé et fonctionnel, nous pouvons passer à l'ajout d'hôtes et de services qui doivent être surveillés.

Cette phase inclut l'ajout des hôtes et des services à surveiller, la création d'utilisateurs, ainsi que l'installation et l'intégration des outils complémentaires tels que NagiosQL, Thruk et NCPA.

L'objectif est de rendre la supervision à la fois fonctionnelle, personnalisée et visuellement exploitable.

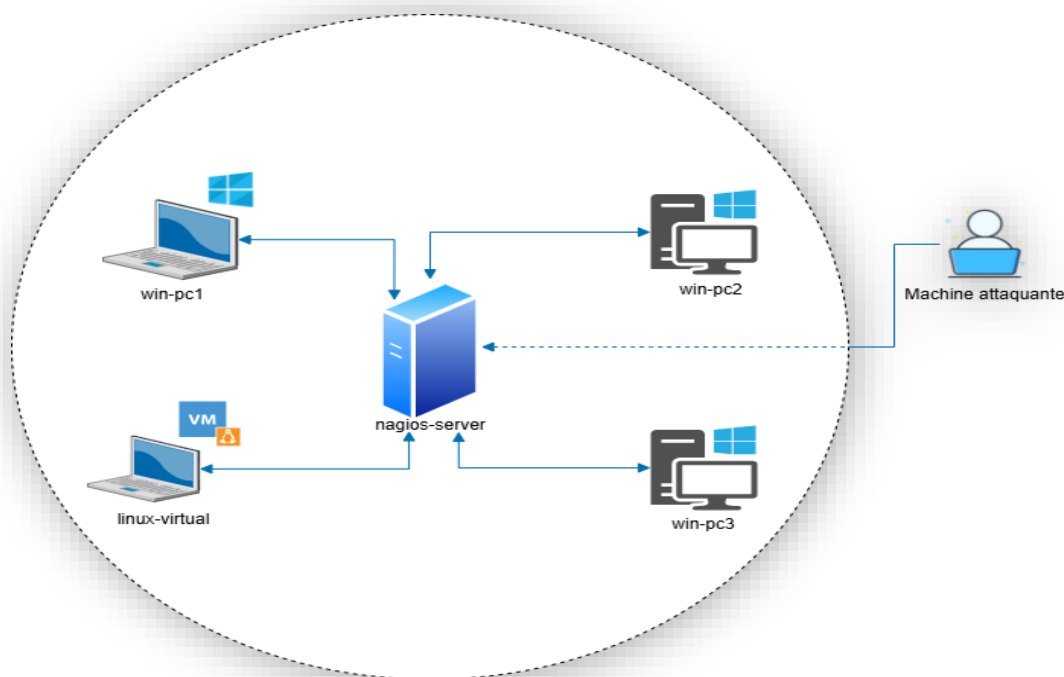


Figure 3.2: Architecture Globale de Supervision

3.5.1 Ajout d'un utilisateur Nagios

L'accès à l'interface web de Nagios est protégé par une authentification, ce qui nécessite la création d'un utilisateur pour y accéder.

Une fois cette étape terminée, nous avons redémarré le service Apache pour appliquer la configuration. À partir de ce moment-là, toute tentative d'accès à l'interface web de Nagios demande l'identifiant et le mot de passe de l'utilisateur que nous avons créé.

3.5.2 Configuration des hôtes

Nous avons mis en place une architecture réseau simple mais représentative, que nous avons ensuite supervisée à l'aide de Nagios. Cette architecture est composée de cinq machines : trois postes Windows connectés au réseau local, un PC Ubuntu installé dans une machine virtuelle VirtualBox, ainsi que le serveur principal Ubuntu sur lequel Nagios est installé. Ce dernier joue le rôle de centre de supervision, chargé de surveiller les autres hôtes.

Chaque hôte supervisé dans notre architecture a été identifié par un nom précis, facilitant ainsi leur gestion et leur suivi dans Nagios. Voici la liste des machines concernées avec une brève description de chacune :

- **nagios-server** : Il s'agit du serveur principal, un ordinateur portable sous Ubuntu 22.04 LTS, sur lequel Nagios est installé et configuré pour assurer la supervision.

- **win-pc1** : Un ordinateur portable fonctionnant sous Windows 11, utilisé comme poste client dans le réseau.
- **linux-virtual** : Une machine virtuelle Ubuntu 22.04 LTS hébergée sur win-pc1 via VirtualBox, simulant un environnement Linux supplémentaire à superviser.
- **win-pc2** : Un ordinateur de bureau sous Windows 11, faisant partie des postes clients supervisés.
- **win-pc3** : Un autre poste de bureau, mais sous Windows 10, complétant ainsi la diversité des systèmes d'exploitation surveillés.

Pour assurer la supervision de nos hôtes distants, **NCPA (Nagios Cross Platform Agent)** a été installé sur chacune de ces machines. Ce choix s'explique par sa simplicité d'installation, sa compatibilité multiplateforme, et surtout sa capacité à exposer des métriques système via une API sécurisée.

N.B : Les étapes détaillées de l'installation de NCPA sur Windows sont disponibles en

Annexe 2 : Installation et Configuration du NCPA

Une fois NCPA installé sur la machine distante, son interface web est accessible à l'URL suivant : `https://<IP_de_l'hôte>:5693`

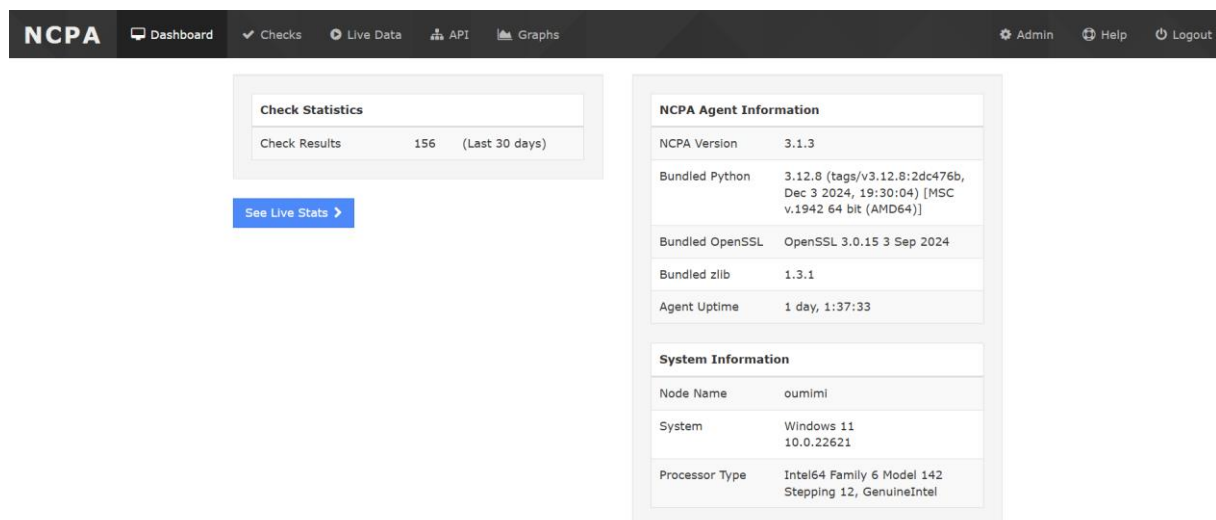


Figure 3.3: Interface NCPA

La mise en place de la supervision commence par la définition des hôtes dans les fichiers de configuration de Nagios. Chaque hôte est identifié par un nom unique, une adresse IP et un alias.

- **Structure de configuration d'un hôte**

Dans notre configuration, les définitions des hôtes sont principalement centralisées dans le répertoire `/usr/local/nagios/etc/hosts/`. Ce dossier est utilisé par NagiosQL pour stocker automatiquement les définitions des hôtes ajoutés via son interface.

En ce qui concerne les définitions de groupes d'hôtes, elles sont par défaut stockées dans les fichiers **localhost.cfg** pour les hôtes Linux, et **windows.cfg** pour les hôtes Windows. Ces fichiers se trouvent dans le répertoire `/usr/local/nagios/etc/objects/`.

Pour que Nagios prenne en compte ces fichiers de configuration spécifiques aux hôtes, il est nécessaire de les référencer dans le fichier principal **nagios.cfg**.

Cette étape est essentielle afin que Nagios puisse charger toutes les définitions des hôtes et services au démarrage.

Chaque ligne `cfg_file` indique à Nagios le chemin du fichier de configuration à charger. Ainsi, toute modification dans ces fichiers sera prise en compte lors du redémarrage De Nagios.

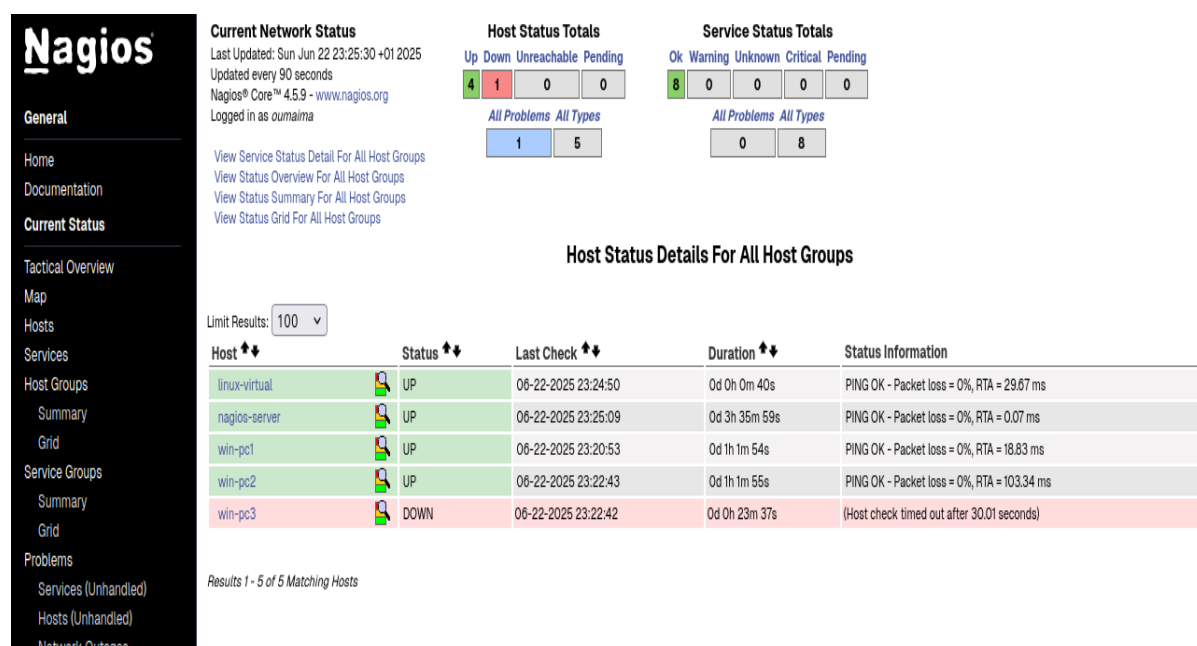


Figure 3.4: Vue d'ensemble des hôtes supervisés dans Nagios

3.5.3 Configuration des Services

Pour chaque service, il faut définir :

- Le nom du service
- L'hôte concerné
- La commande de vérification et ses paramètres (plugin et seuils)

Les définitions des services sont organisées dans des fichiers de configuration situés dans le répertoire `/usr/local/nagios/etc/services/`. Pour que Nagios prenne en compte ces fichiers, nous

avons ajouté `cfg_dir=/usr/local/nagios/etc/services/` dans le fichier principal `nagios.cfg`

La surveillance des services est ensuite accessible en temps réel via l'interface web de Nagios.

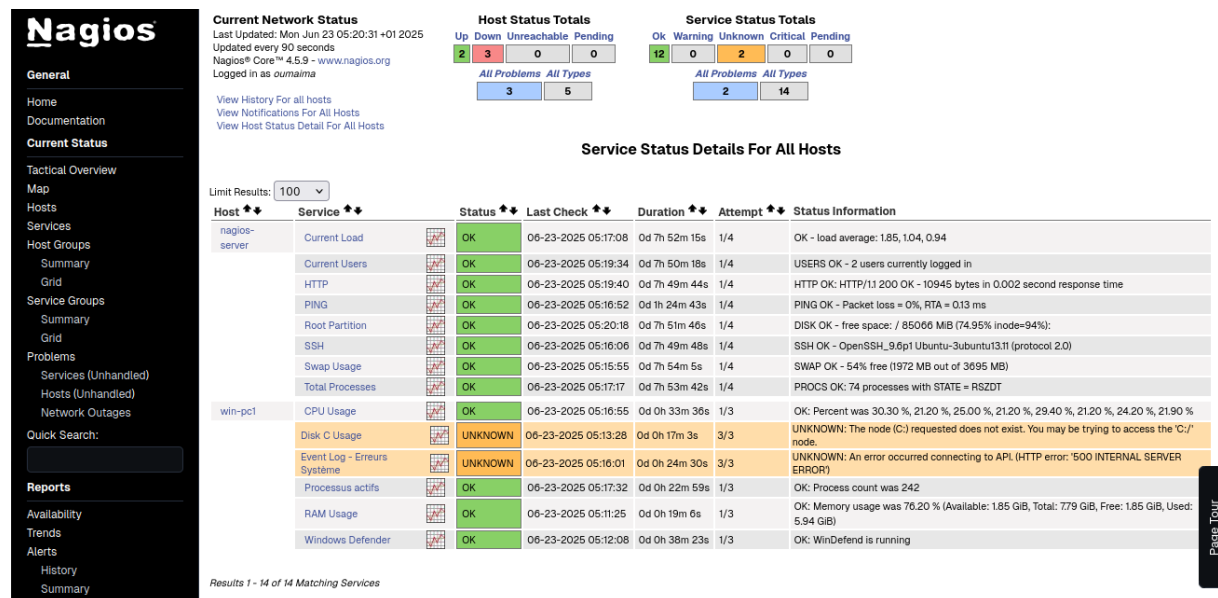


Figure 3.5: Vue d'ensemble des services supervisés dans Nagios

3.5.4 Configuration de l'outil Thruk

Pour améliorer la consultation et la gestion des données de supervision, nous avons configuré l'outil **Thruk**, qui offre une interface web moderne et performante. Thruk communique directement avec Nagios grâce au module **Livestatus**, permettant un accès rapide aux informations sans passer par la lecture des fichiers journaux.

La configuration s'est déroulée en plusieurs étapes clés :

- **Activation du module Livestatus dans Nagios :**
 Le module Livestatus a été compilé et installé, puis intégré dans le fichier `nagios.cfg` via la directive `broker_module`, avec la création d'un socket Unix (`/usr/local/nagios/var/rw/live`) servant de point de communication.
- **Paramétrage de Thruk pour accéder à Livestatus :**
 Le fichier de configuration principal de Thruk (`thruk_local.conf`) a été modifié pour inclure un composant backend pointant vers ce socket, ce qui permet à Thruk de récupérer en temps réel les données Nagios.
- **Configuration du serveur web Apache :**

Apache a été configuré pour héberger l'interface Thruk, en activant les modules nécessaires (notamment CGI), et en définissant les alias pour l'accès via URL.

N.B : Les étapes détaillées de l'installation et configuration du Thruk sont disponibles en Annexe 4 : Installation et Configuration du Thruk

Après ces configurations, Thruk est accessible via l'adresse : « <http://localhost/thruk> »

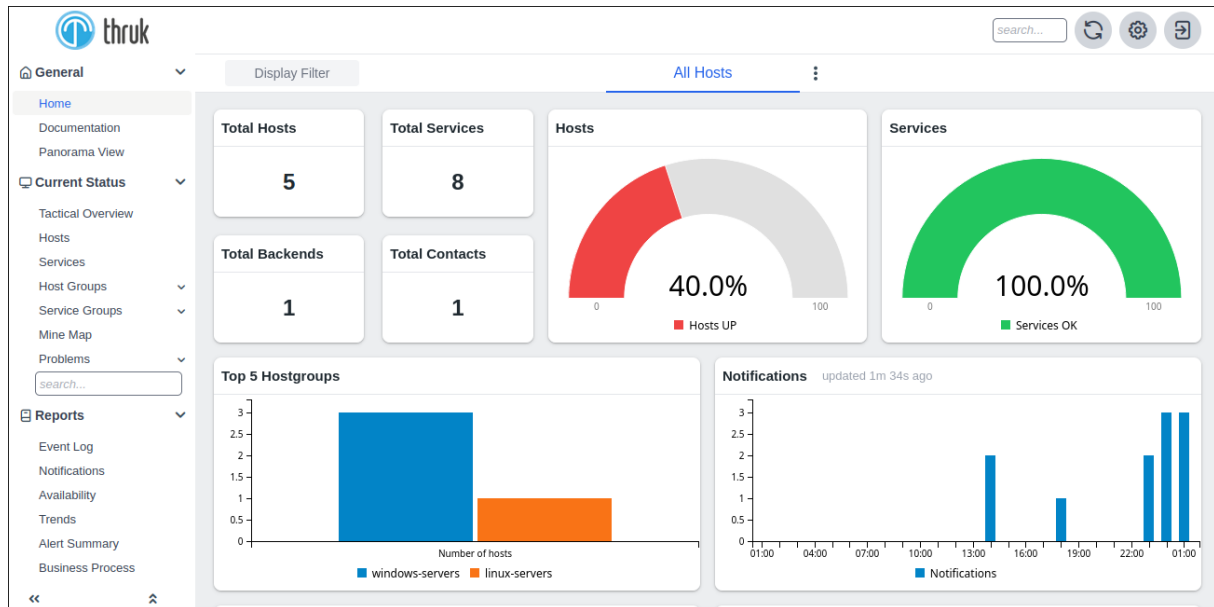


Figure 3.6: Interface web Thruk connectée à Nagios via Livestatus

3.5.5 Configuration de l'outil NagiosQL

Pour moderniser et simplifier la gestion des configurations Nagios, nous avons implémenté NagiosQL, une interface web qui permet d'administrer l'ensemble des éléments de supervision sans manipulation directe des fichiers de configuration.

N.B : Les étapes détaillées de l'installation et configuration du NagiosQL sont disponibles en Annexe 3 : Installation et Configuration du NagiosQL

Après avoir installé NagiosQL. La configuration s'est concentrée sur la liaison entre NagiosQL et Nagios Core.

Pour cela, nous avons paramétré NagiosQL afin qu'il génère et écrive les fichiers de configuration dans des répertoires spécifiques, notamment :

- /usr/local/nagios/etc/hosts/ pour les définitions des hôtes,
- /usr/local/nagios/etc/services/ pour les services.

Ces chemins ont ensuite été intégrés dans le fichier principal de configuration de Nagios (nagios.cfg) par des directives `cfg_dir=`, permettant à Nagios de prendre en compte automatiquement tous les fichiers produits par NagiosQL.

Une fois cette configuration réalisée, nous avons utilisé l'interface web de NagiosQL pour ajouter et modifier les éléments à superviser (hôtes, services, etc.). Après chaque modification,

nous avons utilisé l'option « Écrire les fichiers de configuration » (Write config files) dans NagiosQL pour actualiser les fichiers utilisés par Nagios Core.

Enfin, un redémarrage du service Nagios a permis d'appliquer l'ensemble des changements et d'assurer une supervision fonctionnelle.

Cette approche a permis de simplifier la gestion des configurations et d'éviter les erreurs liées à la modification manuelle des fichiers texte.

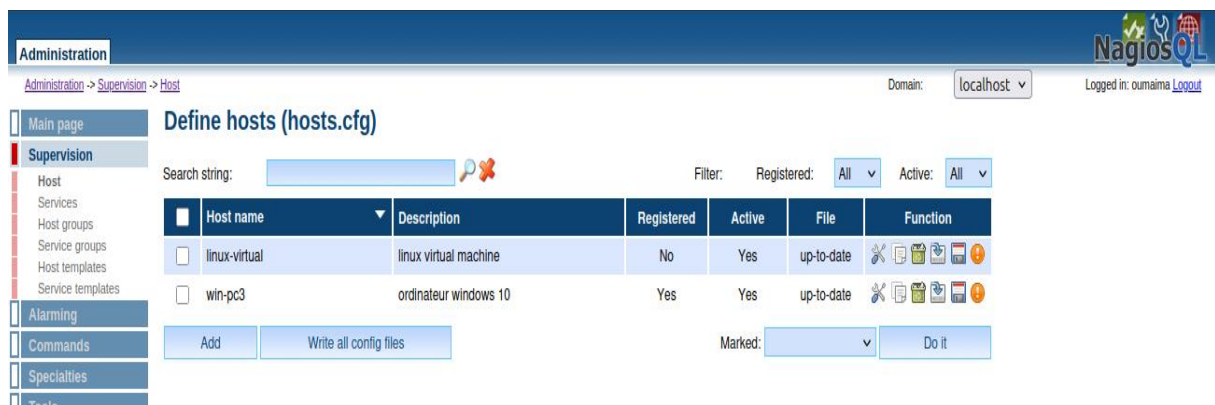


Figure 3.7: Interface web NagiosQL

3.6 Test de sécurité

Après avoir mis en place une infrastructure de supervision complète avec Nagios Core et ses outils complémentaires, il est essentiel de s'assurer que cette infrastructure est capable de résister à d'éventuelles menaces.

Dans cette optique, nous avons réalisé des tests de sécurité ciblés visant à évaluer la robustesse du système supervisé face à deux types d'attaques couramment rencontrées dans les réseaux.

3.6.1 Attaque Brute Force SSH avec Hydra

Objectif :

L'objectif de cette manipulation est de démontrer la faisabilité d'une attaque par force brute sur un service SSH mal protégé. L'attaque a été réalisée depuis une machine Kali Linux ciblant une machine Ubuntu disposant d'un serveur SSH actif.

Identification du service SSH :

Un scan de port a été effectué à l'aide de Nmap afin de vérifier la disponibilité du service SSH (port 22) sur la machine cible. Le port s'est avéré ouvert, ce qui confirme la possibilité d'une connexion distante.


```
(doaa@kalilinux)-[~]
$ nmap -p 22 192.168.100.1
Starting Nmap 7.95 ( https://nmap.org ) at 2025-06-22 18:12 CEST
Nmap scan report for 192.168.100.1
Host is up (0.0010s latency).

PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:D4:E4:77 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.20 seconds
```

Figure 3.8: Commande nmap

Exécution de l'attaque :

L'outil Hydra a été utilisé pour tenter toutes les combinaisons de la wordlist sur le service SSH. L'attaque a permis d'identifier avec succès le mot de passe de l'utilisateur cible (doaa), prouvant ainsi l'efficacité de la méthode face à des mots de passe faibles.

```
(doaa@kalilinux)-[~]
$ hydra -l doaa -P passlist.txt ssh://192.168.100.1
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in
hese *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-06-22 18:
[WARNING] Many SSH configurations limit the number of parallel tasks, it is r
[DATA] max 16 tasks per 1 server, overall 16 tasks, 36 login tries (l:1/p:36)
[DATA] attacking ssh://192.168.100.1:22/
[22][ssh] host: 192.168.100.1 login: doaa password: 10562003
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-06-22 18:
```

Figure 3.9: Commande hydra

Analyse des journaux système :

Une vérification des logs d'authentification sur la machine Ubuntu (/var/log/auth.log) a permis de constater les traces des tentatives de connexion infructueuses ainsi que de la connexion réussie. Ces journaux confirment le déroulement réel de l'attaque.


```

doaa@Doaa:~$ sudo cat /var/log/auth.log | grep "Failed\|Accepted"
[sudo] password for doaa:
2025-06-22T16:02:03.733622+00:00 Doaa dbus-daemon[681]: [system] Failed to activate service 'org.bluez': timed
out (service_start_timeout=25000ms)
2025-06-22T16:30:17.433321+00:00 Doaa sshd[4846]: Failed password for doaa from 192.168.100.2 port 35654 ssh2
2025-06-22T16:30:17.476830+00:00 Doaa sshd[4848]: Failed password for doaa from 192.168.100.2 port 35674 ssh2
2025-06-22T16:30:17.484527+00:00 Doaa sshd[4849]: Failed password for doaa from 192.168.100.2 port 35680 ssh2
2025-06-22T16:30:17.490966+00:00 Doaa sshd[4847]: Failed password for doaa from 192.168.100.2 port 35652 ssh2
2025-06-22T16:30:17.495485+00:00 Doaa sshd[4857]: Failed password for doaa from 192.168.100.2 port 35754 ssh2
2025-06-22T16:30:17.497425+00:00 Doaa sshd[4858]: Failed password for doaa from 192.168.100.2 port 35756 ssh2
2025-06-22T16:30:17.498913+00:00 Doaa sshd[4852]: Failed password for doaa from 192.168.100.2 port 35700 ssh2
2025-06-22T16:30:17.509014+00:00 Doaa sshd[4856]: Failed password for doaa from 192.168.100.2 port 35752 ssh2
2025-06-22T16:30:17.512439+00:00 Doaa sshd[4854]: Failed password for doaa from 192.168.100.2 port 35704 ssh2
2025-06-22T16:30:17.516109+00:00 Doaa sshd[4853]: Failed password for doaa from 192.168.100.2 port 35702 ssh2
2025-06-22T16:30:17.516543+00:00 Doaa sshd[4851]: Failed password for doaa from 192.168.100.2 port 35690 ssh2
2025-06-22T16:30:17.517238+00:00 Doaa sshd[4850]: Failed password for doaa from 192.168.100.2 port 35686 ssh2
2025-06-22T16:30:17.519199+00:00 Doaa sshd[4855]: Failed password for doaa from 192.168.100.2 port 35718 ssh2
2025-06-22T16:30:18.937120+00:00 Doaa sshd[4858]: Accepted password for doaa from 192.168.100.2 port 35756 ssh

```

Figure 3.10: Contenu du auth.log

Cette expérimentation met en évidence les risques critiques liés à l'utilisation de mots de passe faibles pour des services exposés comme SSH. Elle souligne la nécessité d'adopter des mesures de protection adéquates, notamment :

- L'utilisation de mots de passe complexes ;
- La surveillance régulière des journaux système ;
- L'implémentation d'outils de défense comme fail2ban.

3.6.2 Attaque par déni de service (DoS) avec hping3

Objectif :

L'objectif de cette étape est de simuler une attaque par déni de service (DoS) visant le port SSH de la machine cible, afin d'évaluer sa résistance et d'observer l'impact sur les ressources système, notamment via la supervision par Nagios.

Exécution de l'attaque :

L'outil utilisé pour cette attaque est hping3, un générateur de paquets TCP/IP personnalisables. L'attaque a été lancée depuis la machine Kali Linux à l'aide d'une commande qui envoie en continu des paquets TCP SYN vers le port SSH de la machine Ubuntu :

```

(doaa@kalilinux)~$ sudo hping3 -S --flood -p 22 192.168.100.1
HPING 192.168.100.1 (eth0 192.168.100.1): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown

```

Figure 3.11: Commande hping3

Cette commande permet de générer un grand nombre de tentatives de connexion incomplètes (SYN), dans le but de saturer la file d'attente du service SSH et ainsi provoquer une dégradation ou une indisponibilité temporaire.

Effet observé sur le système cible :

Suite à cette attaque, une forte augmentation de la consommation CPU a été observée sur la machine Ubuntu. Cela s'explique par la charge générée par la gestion de milliers de paquets réseau entrants.

Nagios, configuré pour surveiller l'utilisation du processeur de la machine cible, a détecté cette montée en charge. Le service CPU est passé en état warning, signalant ainsi une surcharge inhabituelle du système.

Cela confirme que le système de supervision permet bien de détecter en temps réel les effets d'une attaque réseau.

CPU avant l'attaque DoS :

CPU Usage RealTime	OK	06-22-2025 18:13:38	0d 0h 3m 8s	1/3	OK - CPU Usage is 4.3%
--------------------	----	---------------------	-------------	-----	------------------------

CPU après l'attaque DoS :

CPU Usage RealTime	WARNING	06-22-2025 18:01:31	0d 0h 0m 26s	1/3	WARNING - CPU Usage is 78.3%
--------------------	---------	---------------------	--------------	-----	------------------------------

Cette simulation démontre l'impact réel qu'une attaque DoS peut avoir sur la disponibilité et la performance d'un service, même sans provoquer une panne totale. Pour limiter les risques à l'avenir, il est fortement recommandé de :

- Mettre en place des règles de pare-feu efficaces (iptables)
- Utiliser des outils de protection automatisés comme fail2ban
- Assurer une supervision continue des ressources critiques

Conclusion

Bien que Nagios soit un outil puissant et largement reconnu pour la supervision des systèmes et des réseaux, son interface par défaut peut s'avérer peu intuitive, notamment pour les utilisateurs non techniques. Dans un environnement professionnel où coexistent différents profils, techniciens, responsables métiers ou utilisateurs sans expertise informatique, il est essentiel de proposer une interface claire, accessible et orientée utilisateur.

C'est dans cette optique que nous avons entrepris le développement d'une interface web personnalisée, plus conviviale et adaptée à un usage quotidien par des profils variés. Le chapitre suivant sera ainsi consacré à la présentation de cette interface, en détaillant son architecture, ses fonctionnalités et les choix techniques qui ont guidé sa réalisation.

4 Conception et développement de la plateforme de supervision

Introduction

Ce chapitre présente la phase de conception et de développement de la plateforme de supervision que nous avons réalisée dans le cadre de notre projet. Après avoir installé et configuré l'environnement de supervision avec Nagios et ses composants, nous avons conçu une interface utilisateur dédiée à la visualisation des données collectées.

Dans ce chapitre, nous détaillerons les outils et technologies utilisés, la modélisation UML de notre système, ainsi que le design de l'interface. L'objectif est de décrire l'architecture logicielle que nous avons adoptée, les choix techniques effectués, et les étapes de développement qui ont permis de concrétiser la plateforme.

4.1 Les outils et technologies utilisés

Pour le développement de notre plateforme de supervision, nous avons eu recours à plusieurs outils et technologies, choisis en fonction de leur compatibilité, de leur stabilité, et de leur capacité à répondre aux exigences du projet. Ces outils couvrent à la fois la supervision réseau, le développement de l'interface, ainsi que la communication entre les différentes couches du système.

Voici les principaux éléments utilisés :

➤ Nagios Core :

Nagios, ou Nagios Core est un logiciel ordonnanceur qui surveille les systèmes, les réseaux et l'infrastructure. Nagios offre des services de surveillance et d'alerte pour les serveurs, les commutateurs, les applications et les services. Il alerte les utilisateurs en cas d'incidents et les avertit une deuxième fois lorsque le problème a été résolu. Nagios a été conçu à l'origine pour fonctionner sous Linux, mais il fonctionne aussi bien sur d'autres variantes d'Unix.



Figure 4.1: Nagios

➤ **HTML :**

HTML, est un langage informatique pour rédiger des pages web. Grâce à lui il est possible de rédiger de l'hypertexte, de mettre en forme le contenu, de faire des formulaires de saisie, de rajouter dans la page des images, vidéos ou des graphismes ou encore de faire la sémantique de la page web. Ce langage fonctionne avec un système de balises qui vont servir à mettre en avant les différents éléments grâce à des titres, des sous-titres, etc.



Figure 4.2: HTML

➤ **CSS :**

CSS est un langage utilisé pour décrire l'apparence et la mise en forme d'un document écrit en HTML ou XML. En d'autres termes, alors que le HTML fournit la structure de la page (comme les titres, les paragraphes et les liens), le CSS détermine comment ces éléments doivent apparaître à l'écran. Cela inclut des aspects tels que la couleur, la taille de la police, l'espacement, et même des animations.



Figure 4.3: CSS

➤ **JavaScript:**

JavaScript est un langage de programmation utilisé par les développeurs pour concevoir des sites web interactifs. Les fonctions JavaScript peuvent permettre d'améliorer l'expérience utilisateur d'un site web, de la mise à jour des flux de médias sociaux à l'affichage d'animations et de cartes interactives. En tant que langage de script côté client, c'est l'une des principales technologies du web.



Figure 4.4: JavaScript

➤ **PHP :**

Le PHP, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.



Figure 4.5: PHP

➤ **MYSQL:**

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.



Figure 4.6: MYSQL

➤ **MK Livestatut :**

Livestatus est l'interface la plus importante de Checkmk.

Il s'agit du moyen le plus rapide d'obtenir toutes les données des ordinateurs hôtes et des services supervisés, y compris les données en temps réel. Ainsi, les données de l'aperçu sont par exemple récupérées directement via cette interface. Comme elles sont lues directement depuis la RAM, cela évite les accès lents au disque dur et permet d'accéder rapidement à la supervision sans trop solliciter le système. Les données sont organisées en tables et colonnes afin de les structurer.



Figure 4.7: MK Livestatut

4.2 Diagrammes UML

Afin d'assurer une organisation cohérente et efficace du développement de notre plateforme de supervision, nous avons eu recours à la modélisation UML (Unified Modeling Language). Cette étape nous a permis de représenter les principales entités du système, leurs responsabilités, ainsi que les interactions entre elles. L'objectif de cette modélisation est de clarifier l'architecture logicielle avant le passage à l'implémentation.

4.2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation suivant illustre les interactions possibles entre l'utilisateur et la plateforme de supervision. Il représente les différentes fonctionnalités offertes par le système, regroupées autour des actions que peut réaliser un utilisateur .

Ce diagramme permet de visualiser les besoins fonctionnels du système du point de vue de l'utilisateur, et sert de base pour la conception des interfaces et des scénarios d'utilisation.

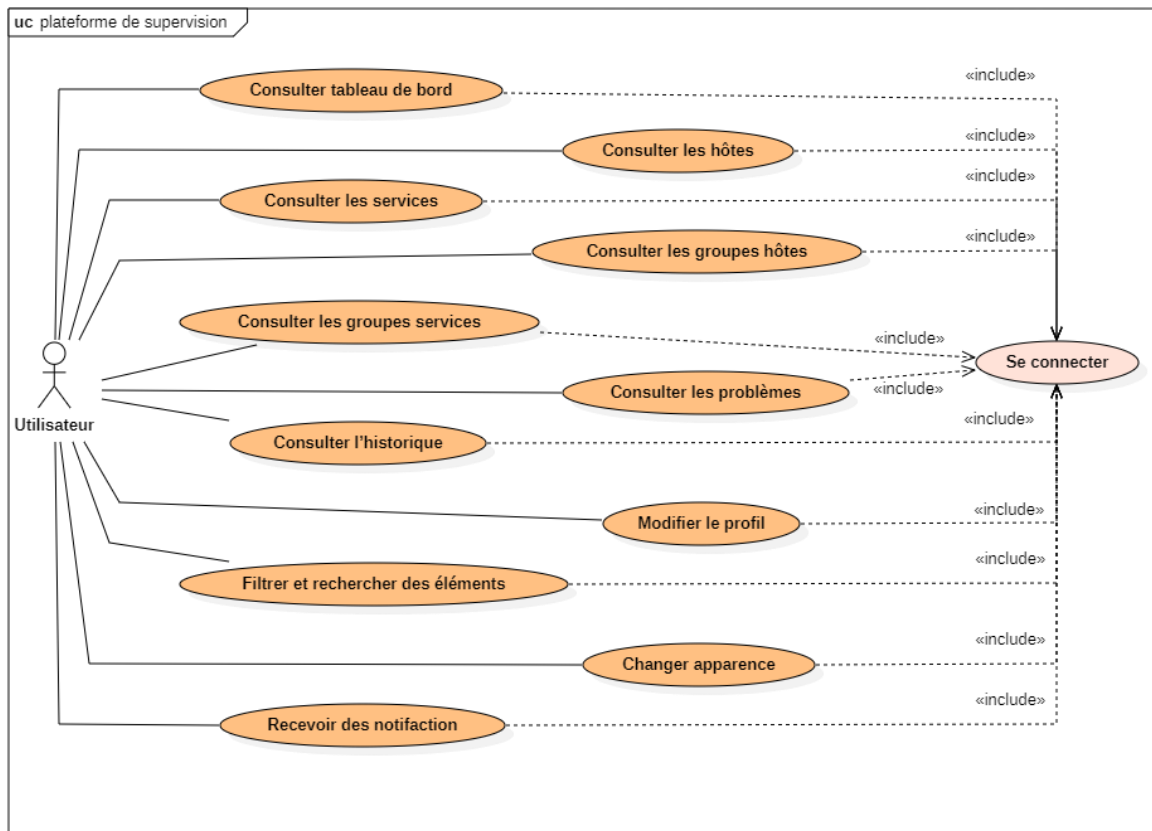


Figure 4.8: Diagramme de cas d'utilisation

4.2.1.1 Description textuelle du cas d'utilisation :

Cette partie décrit le cas d'utilisation « Consulter les hôtes », choisi comme exemple pour illustrer une action courante sur l'interface de supervision.

- **Identification :**

Nom du cas : Consulter les hôtes

Objectif : Permettre à l'utilisateur de consulter dynamiquement la liste des hôtes supervisés, leur état, et leurs détails à jour.

Acteurs : Utilisateur (authenticifié)

Date : 06/2025

Responsable : Équipe Nagios Web

- **Séquencement :**

Le cas d'utilisation commence lorsqu'un utilisateur authenticifié souhaite consulter les hôtes supervisés via l'interface.

Pré-conditions :

- L'utilisateur est connecté à la plateforme.
- Le service Nagios et le socket Livestatus sont disponibles.

Enchaînement nominal :

1. L'utilisateur accède à la section « Hôtes » via le menu principal.
2. Le navigateur exécute automatiquement une requête AJAX vers l'API /api/hosts.php.
3. Le backend PHP envoie une requête au socket Livestatus pour récupérer les informations des hôtes.
4. Le système reçoit les données (nom, IP, groupe, état, etc.) et les formate en JSON.
5. Le navigateur interprète ces données et remplit dynamiquement le tableau HTML.
6. L'utilisateur visualise les hôtes, peut les trier, filtrer ou rechercher des noms spécifiques.

Enchaînements alternatifs

- En (2) : si la requête AJAX échoue (serveur indisponible), un message d'erreur s'affiche : « Erreur de chargement des données. »
- En (3) : si le socket Nagios est inactif ou inaccessible, le backend retourne une erreur serveur (500) avec un message explicite.
- En (6) : si aucun hôte ne correspond au filtre ou à la recherche, un message comme « Aucun résultat trouvé » s'affiche.

Post-conditions

- La liste des hôtes est affichée correctement sur l'interface utilisateur.
- L'utilisateur peut consulter l'état en temps réel des machines supervisées.

4.2.2 Diagramme de séquence

Les diagrammes de séquence suivants illustrent le déroulement temporel des interactions entre les différents composants du système lors de deux fonctionnalités clés. Ils permettent de visualiser les échanges de messages entre les acteurs (utilisateur, navigateur, interface web, API) et les composants internes (services, base de données, socket Livestatus) lors de l'authentification d'un utilisateur et de la récupération des hôtes supervisés.

Ces diagrammes sont essentiels pour comprendre la logique d'exécution du système et pour valider la cohérence des échanges entre les couches de l'architecture.

- **Authentification :**

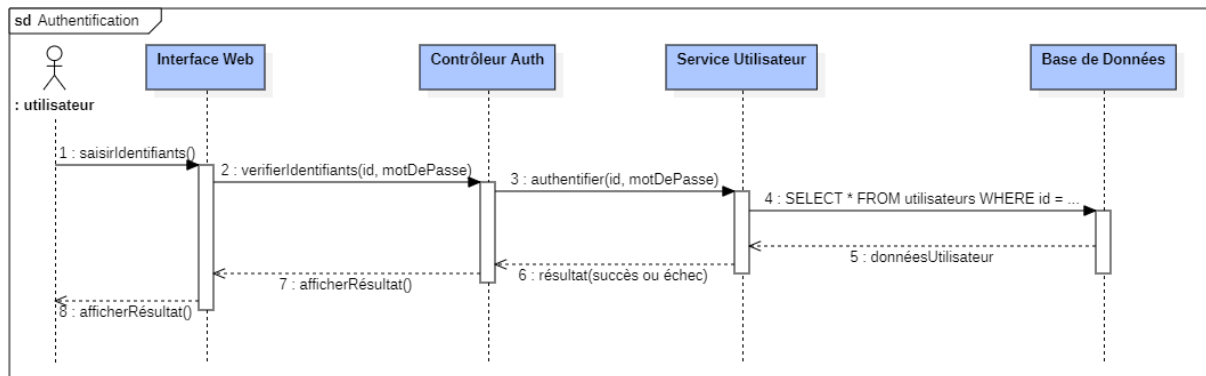


Figure 4.9: Diagramme de séquence authentification

- **Consultation des services :**

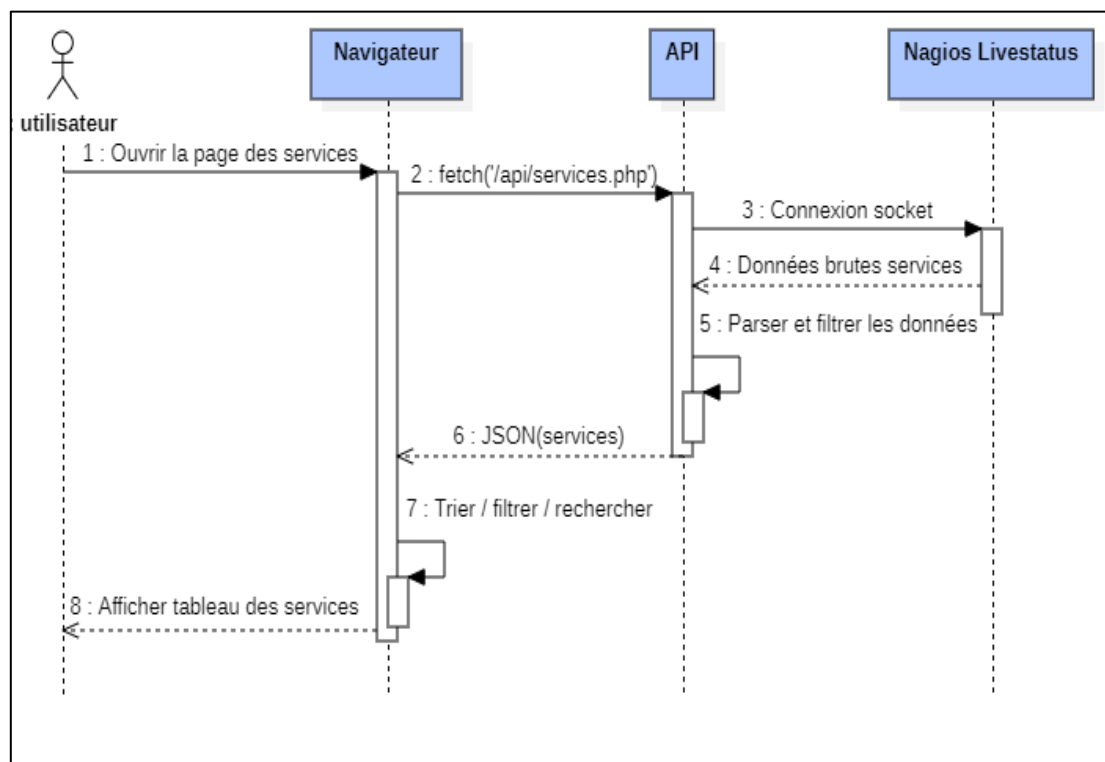


Figure 4.10: Diagramme de séquence consultation des services

4.3 Design de l'interface

4.3.1 Organisation générale de l'interface

L'interface utilisateur de la plateforme de supervision a été conçue de manière structurée afin de faciliter la lecture des informations essentielles et la navigation entre les différentes fonctionnalités. Elle est divisée en trois grandes zones principales, clairement identifiables :

- **Le menu latéral gauche** : toujours visible, permet un accès rapide aux principales sections de la plateforme : Tableau de bord, Hôtes, Services, Groupes Hôtes, Groupes Services, Problèmes, Historique et Profil. Ce menu assure une navigation fluide et intuitive entre les différentes pages sans rechargement complet du site.
- **La barre supérieure** : discrète mais fonctionnelle, contient un champ de recherche pour retrouver un hôte ou un service... rapidement, une icône de notification, ainsi qu'un bouton d'apparence permettant de basculer entre le mode clair et le mode sombre de l'interface et un bouton de déconnexion.
- **La zone centrale de contenu** : qui constitue le cœur de l'interface, affiche dynamiquement les données supervisées en fonction de la section sélectionnée.

4.3.2 Design des pages

Chaque page de l'interface a été conçue avec une attention particulière à la clarté visuelle, à l'organisation logique de l'information, et à la facilité d'accès aux données critiques. Le design adopte une approche minimaliste mais efficace, où la priorité est donnée à la lisibilité et à la compréhension rapide de l'état du système.

➤ Page authentication

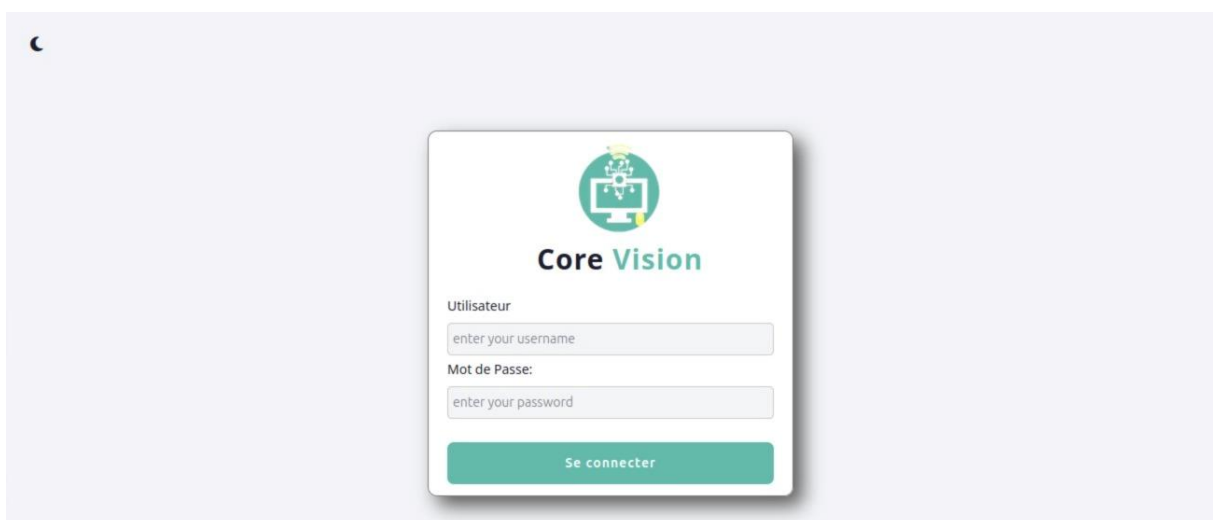


Figure 4.11: Page authentication

Cette page permet à l'utilisateur de s'authentifier avant d'accéder à la plateforme. Elle contient deux champs de saisie pour le nom d'utilisateur et le mot de passe, ainsi qu'un bouton *Se connecter*. L'interface est simple, centrée et épurée, avec un logo symbolisant la supervision intelligente.

➤ Page d'accueil (tableau de bord)

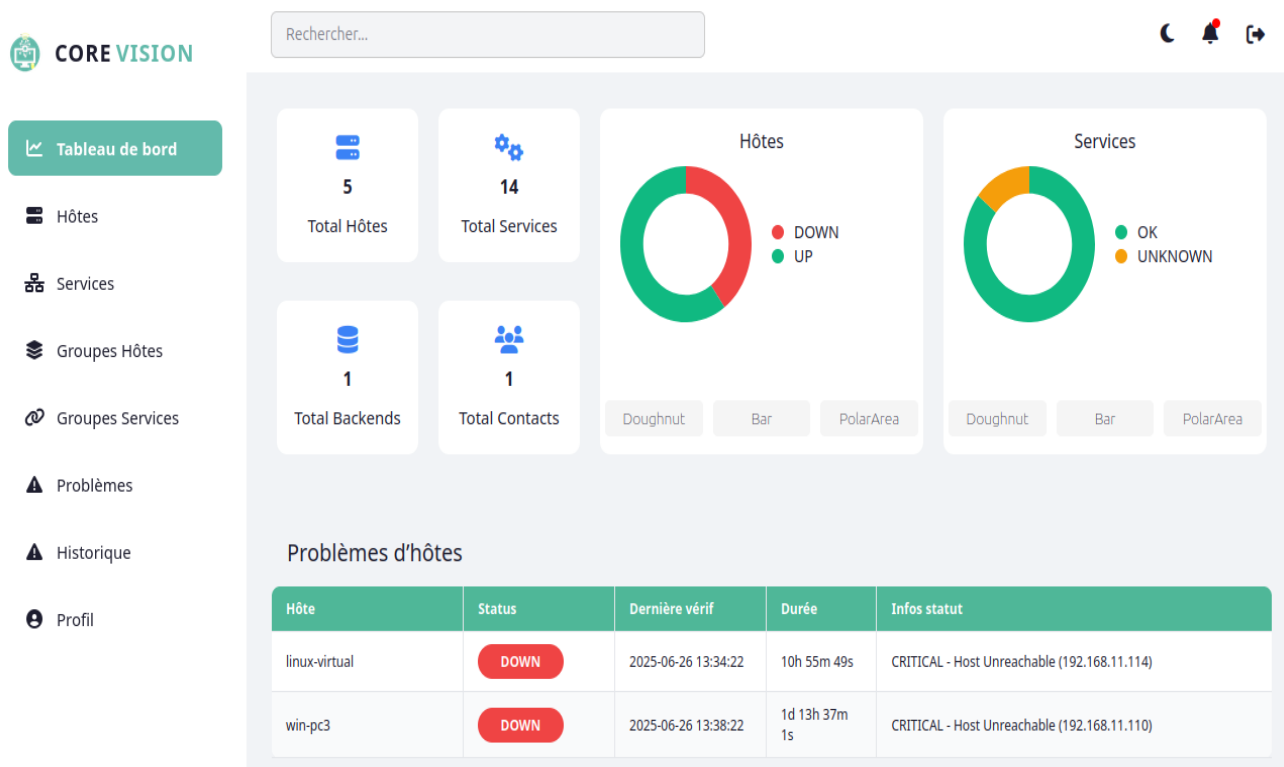


Figure 4.12: Page d'accueil (tableau de bord)

Cette page présente un résumé global de l'état du système supervisé. On y trouve :

- Des compteurs indiquant le nombre total d'hôtes, de services, de backends et de contacts.
- Des graphiques circulaires montrant la répartition des statuts (OK, Warning, Critical, Unknown).
- Des tableaux listant les problèmes de services et hôtes actifs, avec leur nom, état, durée, et message d'erreur...
- Un bouton « Voir plus » permet d'accéder au détail complet des incidents.

➤ Pages Hôtes, Services et Groupes :

✓ En-tête d'état global (section supérieure) :



Figure 4.13: En-tête d'état global

Les pages « Hôtes », « Services », « Groupes hôtes » et « Groupes services » commencent par une section récapitulative affichant les indicateurs globaux du système. Elle inclut :

- Un graphique circulaire représentant la répartition des hôtes selon leur état : UP, DOWN, UNREACHABLE, PENDING.
- Des compteurs par statut, accompagnés d'icônes et de couleurs (vert, rouge, orange, jaune).
- Une barre de santé du réseau, affichant le pourcentage d'hôtes et de services en bon état.
- Le temps moyen de réponse, calculé sur les dernières vérifications.
- Un rappel du nombre total d'hôtes et de services supervisés.

Cette section permet à l'utilisateur d'avoir un aperçu rapide et visuel de l'état général avant de descendre dans le détail du tableau.

✓ Fonctions de filtrage et de recherche :



Figure 4.14: Fonctions de filtrage et de recherche

Une barre d'outils est présente au-dessus des tableaux dans les pages Hôtes, Services, Groupes hôtes et Groupes services. Elle permet à l'utilisateur d'interagir avec les données affichées grâce à :

- Un bouton « **Filtres** » pour afficher uniquement les éléments selon leur état (UP, DOWN, etc.).
- Un bouton « **Réinitialiser le tri** » pour revenir à l'ordre d'affichage par défaut.
- Un **champ de recherche** pour retrouver rapidement un hôte, un service ou des groupes par son nom.

Ces outils rendent la navigation dans les tableaux plus efficace, en facilitant l'identification d'éléments précis ou d'anomalies dans un grand volume de données.

✓ Tableaux Hôtes et services :

Host	Adresse IP	État	Groupe	Statut Info	Dernière vérif
linux-virtual	192.168.11.114	DOWN	-	Injoignable	27/06/2025 09:45:42
nagios-server	127.0.0.1	UP	linux-servers	Fonctionnel	27/06/2025 09:46:42
win-pc1	192.168.11.101	UP	windows-servers	Fonctionnel	27/06/2025 09:47:42
win-pc2	192.168.11.103	DOWN	windows-servers	Injoignable	27/06/2025 09:48:13
win-pc3	192.168.11.110	DOWN	windows-servers	Injoignable	27/06/2025 09:49:42

Figure 4.15: Tableaux hôtes

Host	Service	État	Groupe	Statut Info	Dernière vérif
nagios-server	Current Load	OK	-	Fonctionnel	27/06/2025 10:48:15
	Current Users	OK	-	Fonctionnel	27/06/2025 10:49:11
	HTTP	OK	-	Fonctionnel	27/06/2025 10:47:51
	PING	OK	-	Fonctionnel	27/06/2025 10:48:45
	Root Partition	OK	-	Fonctionnel	27/06/2025 10:50:42

Figure 4.16: Tableaux services

Les pages Hôtes et Services reposent sur des tableaux dynamiques qui affichent les données collectées en temps réel.

- Dans la page Hôtes, chaque ligne du tableau correspond à un hôte supervisé, avec des colonnes indiquant son nom, son adresse IP, son état, son groupe d'appartenance, un statut textuel (ex. : "fonctionnel") et la dernière vérification effectuée.
- Dans la page Services, les services sont affichés ligne par ligne avec leur état (OK, WARNING, etc.), le groupe associé, un message de statut, la dernière vérification effectuée.

✓ Tableaux Groupes hôtes et services :

Groupe	Hôte	Statut du Hôte	Nb. de Services	Statut des Services	Détails
linux-servers	nagios-server	UP	8	8 OK	
windows-servers	win-pc1	UP	6	4 OK 2 UNKNOWN	
	win-pc2	DOWN	0	-	
	win-pc3	DOWN	0	-	

Figure 4.17: Tableaux groupes hôtes

Groupe	Service	Statut du Service	Détail
windows-services	Disk C Usage	UNKNOWN	
	Event Log - Erreurs Système	UNKNOWN	
	RAM Usage	OK	
	Windows Defender	OK	

Figure 4.18: Tableaux groupes services

Les pages « Groupes Hôtes » et « Groupes Services » affichent une vue agrégée des entités supervisées par groupe.

- Dans la page Groupes Hôtes, les hôtes sont listés par catégories fonctionnelles leur état global, le nombre de services, et un résumé du statut des services (nombre de OK, WARNING, CRITICAL...), et une icône qui permet d'accéder à plus de détails.
- Presque la même chose pour la page Groupes Services, on y retrouve les services listés par catégories fonctionnelles, le nom du service, son état (OK, CRITICAL...), et une icône qui permet d'accéder à plus de détails.

Ces vues regroupées facilitent l'analyse par ensemble logique (infrastructure, applications, etc.) et permettent une priorisation plus rapide des incidents.

➤ Page Problèmes :

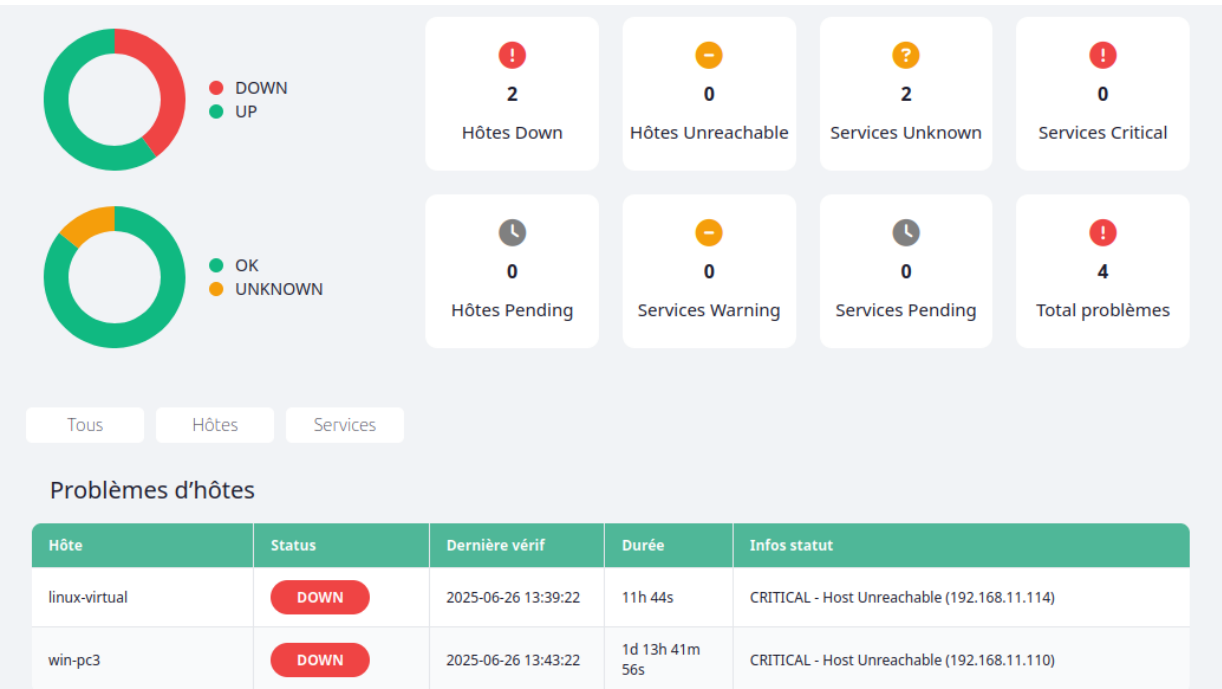


Figure 4.19: Page Problèmes

La page « Problèmes » regroupe l'ensemble des anomalies détectées sur les hôtes et services. Elle présente :

- Deux graphes circulaires montrant la répartition des statuts Hôtes (UP/DOWN) et Services (OK, CRITICAL, UNKNOWN, PENDING, UNREACHABLE).
- Des compteurs affichant le nombre d'hôtes down, unreachable, pending, ainsi que les services en état critical, warning, unknown ou pending.
- Un résumé du nombre total de problèmes actifs.
- Des onglets de filtre pour afficher uniquement les problèmes liés aux hôtes ou aux services.
- Des tableaux listant les hôtes ou services en erreur, avec leur état, la date de dernière vérification, la durée du problème, et un message explicatif (ex. : timeout de vérification).

Cette page permet à l'utilisateur de se concentrer uniquement sur les éléments critiques à corriger.

➤ Page Historique :



Figure 4.20: Page Historique

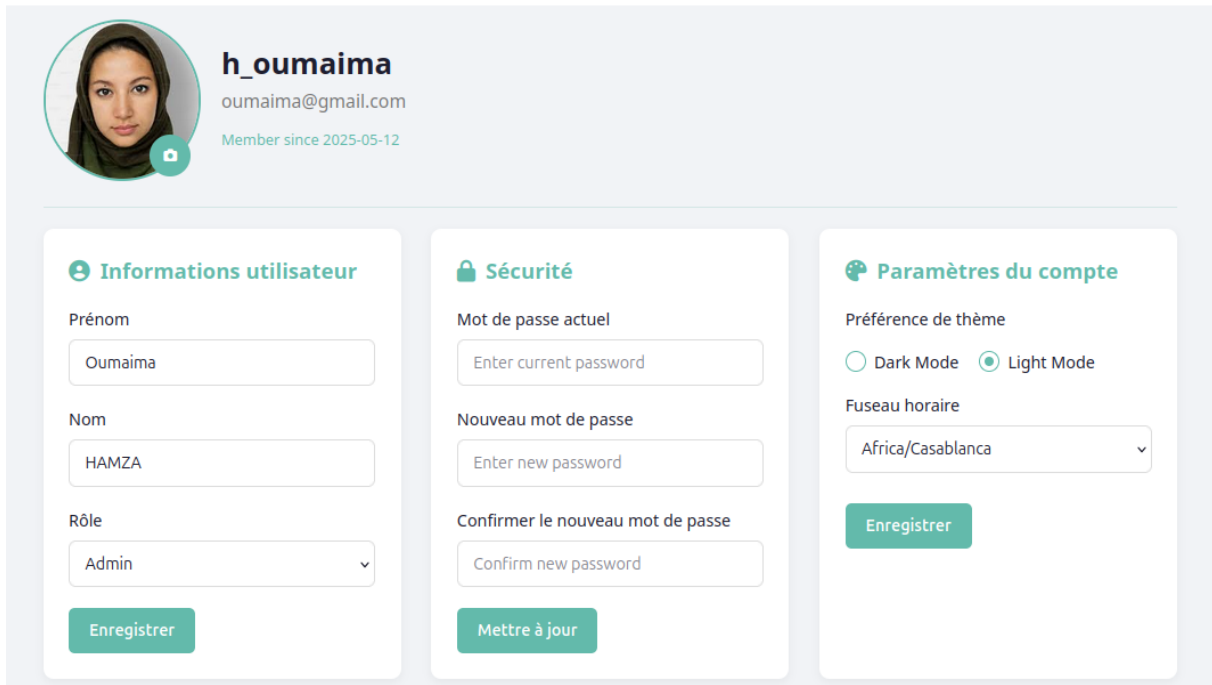
La page Historique est une interface qui permet de visualiser l'ensemble des alertes et événements enregistrés par Nagios, classés par date et par heure. Elle offre une vue centralisée des statuts des hôtes et des services surveillés.

La page est organisée en plusieurs sections claires :

- En-tête : un titre explicite et l'heure de la dernière actualisation des données. Un sélecteur permet de filtrer les alertes par période (1h, 6h, 24h, 7j ou tout l'historique), offrant une vue adaptée aux besoins de l'utilisateur. Un bouton de rafraîchissement manuel est disponible pour mettre à jour les données en temps réel.

- Corps principal : L'interface présente les alertes classées du plus récent au plus ancien, permettant de voir immédiatement les derniers événements. Les alertes sont regroupées par jour, avec une séparation nette entre les différentes dates pour une lecture claire. Chaque alerte affiche précisément
 - L'heure exacte de détection
 - Le type (host ou service)
 - L'élément concerné (nom du serveur ou du service)
 - Un statut visuel coloré.
 - La mention SOFT (alerte non confirmée) ou HARD (problème avéré)

➤ Page Profil :



The screenshot displays a user profile page for 'h_oumima' (email: oumima@gmail.com, member since 2025-05-12). The page is divided into three main sections:

- Informations utilisateur:** Contains fields for 'Prénom' (Oumima), 'Nom' (HAMZA), and 'Rôle' (Admin). A green 'Enregistrer' button is at the bottom.
- Sécurité:** Contains fields for 'Mot de passe actuel' (Enter current password), 'Nouveau mot de passe' (Enter new password), and 'Confirmer le nouveau mot de passe' (Confirm new password). A green 'Mettre à jour' button is at the bottom.
- Paramètres du compte:** Contains 'Préférence de thème' (Dark Mode / Light Mode, with Light Mode selected) and 'Fuseau horaire' (Africa/Casablanca). A green 'Enregistrer' button is at the bottom.

Figure 4.21: Page Profil

La page « Profil » permet à l'utilisateur de consulter et modifier ses informations personnelles. Elle est divisée en trois sections principales :

- **Informations utilisateur :** modification du nom, rôle, ou identifiant visible.
- **Sécurité :** changement de mot de passe via un formulaire sécurisé.
- **Paramètres du compte :** choix du thème d'apparence (mode clair ou sombre) et du fuseau horaire.

La présentation est claire, avec des champs bien espacés, des boutons explicites (Enregistrer, Mettre à jour), et une séparation logique des blocs. Cette page améliore l'expérience utilisateur en offrant une personnalisation basique mais utile.

4.3.3 Dynamisme et échanges avec le backend

L'interface de la plateforme est conçue pour interagir de manière dynamique avec le backend, sans rechargement complet des pages. Ce dynamisme repose principalement sur l'utilisation de **JavaScript** et de **requêtes AJAX** envoyées vers des scripts PHP spécifiques.

Lorsqu'un utilisateur navigue sur l'interface, par exemple en accédant à la page « Hôtes », une requête AJAX est automatiquement envoyée vers le fichier `hosts.php` situé dans le dossier `/api/`. Ce script agit comme une API, et son rôle est de :

- **Interroger le socket Livestatus** de Nagios pour obtenir les données en temps réel (états des hôtes, adresses IP, groupes, etc.).
- **Traiter et formater ces données** au format JSON.
- **Renvoyer la réponse au navigateur**, qui se charge ensuite d'afficher les données dans un tableau dynamique.

Ce mécanisme permet à l'utilisateur de consulter des informations actualisées sans devoir recharger la page. De plus, il peut effectuer des actions comme :

- Filtrer les résultats (ex. : afficher uniquement les hôtes en panne),
- Effectuer des recherches en temps réel,
- Trier les colonnes (par nom, état, groupe...)

Le même principe est utilisé pour les autres pages. En cas d'erreur côté serveur ou si le socket Livestatus est indisponible, des messages d'erreur explicites sont prévus (ex. : "Erreur de chargement des données").

Ce système d'échange permet donc à l'interface d'être **légère, fluide et réactive**, tout en assurant une communication directe avec le moteur de supervision Nagios.

4.3.4 Limites actuelles et pistes d'amélioration

Bien que l'interface actuelle de la plateforme de supervision soit fonctionnelle, intuitive et visuellement claire, elle présente certaines limites qui peuvent être améliorées dans les versions futures :

- **Absence de modification ou d'écriture des données**

L'interface actuelle est principalement en lecture seule. L'utilisateur peut consulter l'état des hôtes et services, mais ne peut pas modifier directement les configurations ou interagir de manière active avec les éléments supervisés (ajout/suppression d'hôtes, relance de service, commentaires, etc.). Intégrer une couche d'administration (avec sécurité renforcée) permettrait d'élargir les possibilités d'interaction avec Nagios.

- **Pas de gestion avancée des utilisateurs**

La gestion des utilisateurs reste basique : il n'existe pas de rôles (admin, opérateur, invité), ni de système de permissions détaillées. Une amélioration possible serait l'intégration d'un système d'authentification étendu, avec différents niveaux d'accès et une interface dédiée à la gestion des comptes.

- **Interface non compatible avec les appareils mobiles**

L'interface actuelle a été conçue uniquement pour un usage sur ordinateur de bureau. Elle ne prend pas en charge l'affichage sur des écrans de petite taille, comme ceux des smartphones ou des tablettes, ce qui limite l'accessibilité de la plateforme à un seul type de support.

Conclusion

Ce chapitre a présenté en détail la conception technique et l'implémentation de notre plateforme de supervision, construite autour de Nagios. Nous avons décrit les outils et technologies utilisés, la modélisation UML du système, ainsi que l'architecture et le design de l'interface utilisateur. Cette interface, moderne et dynamique, vise à rendre la supervision accessible, claire et réactive, même pour les utilisateurs non techniques.

Malgré ses fonctionnalités avancées, certaines limites subsistent, notamment l'absence de modification directe des configurations, la gestion simplifiée des utilisateurs, et l'absence de version mobile. Ces points ouvrent des perspectives d'amélioration pour les évolutions futures de la plateforme.

Conclusion Générale

La réalisation de ce projet de fin d'études nous a permis d'approfondir nos compétences à la fois techniques et méthodologiques dans le domaine de la supervision réseau. À travers le déploiement d'une solution basée sur Nagios et l'élaboration d'une interface personnalisée, nous avons répondu à une problématique concrète : rendre la supervision plus accessible, efficace et adaptée aux besoins actuels des organisations.

Nous avons pu explorer en détail les principes de fonctionnement de Nagios, les protocoles de supervision tels que SNMP, ainsi que les outils d'extension permettant une meilleure visualisation et gestion du système supervisé. Le développement de notre propre interface, dynamique et responsive, constitue une réelle valeur ajoutée au système en offrant une expérience utilisateur simplifiée.

Ce projet représente une base solide pour de futures améliorations, telles que l'ajout de fonctionnalités d'administration, la gestion des utilisateurs avec différents niveaux d'accès, ou encore l'adaptation de l'interface aux supports mobiles. Il ouvre également la voie à une intégration dans des environnements professionnels réels, répondant aux enjeux de fiabilité, de sécurité et d'efficacité dans la gestion des infrastructures informatiques.

Bibliographie

- [1.01] Wojciech Kocjan, Piotr Beltowski - Learning Nagios-Packt Publishing (2016)
- [3.01] <https://support.nagios.com/kb/category.php?id=62>
- [3.03] <https://www.websentra.com/nagios-core-beginners-guide/>
- [3.05] <https://www.networkmanagementsoftware.com/nagios-core-beginners-guide/>
- [3.05] <https://library.nagios.com/documentation/step-by-step-guide-installing-and-monitoring-windows-10-with-nagios-cross-platform-agentncpa/>
- [3.05] <https://assets.nagios.com/downloads/ncpa/docs/Installing-NCPA.pdf>
- [3.05] <https://www.smnet.fr/icinga/icinga-nagiosql.html>
- [3.05] <https://fr.scribd.com/document/139681336/Nagiosql-3-2-Installation>
- [3.05] <https://community.spiceworks.com/t/setup-nagiosql/1012302>
- [3.05] <https://www.thruk.org/documentation/install.html>
- [3.05] <https://www.thruk.org/documentation/configuration.html>
- [3.05] <https://github.com/sni/Thruk/issues/1387>
- [3.05] <https://serverfault.com/questions/952199/omd-thruk-doesnt-detect-nagios-config>
- [3.05] <https://support.nagios.com/forum/>
- [4.02] <https://www.lije-creative.com/tuto-application-php-api-json/>
- [4.02] <https://web.iamvdo.me/js/ajax/>

Annexes

Dans le cadre de ce projet, plusieurs outils de supervision ont été installés, configurés et intégrés afin de mettre en place une plateforme complète, stable et fonctionnelle. Les étapes techniques nécessaires à cette mise en œuvre sont nombreuses, parfois complexes, et ont été volontairement synthétisées dans le corps principal du rapport pour en faciliter la lecture.

Les présentes annexes ont pour objectif de documenter en détail l'ensemble des procédures de configuration et d'installation réalisées tout au long du projet. Elles permettent ainsi de garder une trace précise des manipulations effectuées et d'offrir au lecteur un support technique complet pour reproduire l'environnement mis en place.

Chaque annexe traite spécifiquement d'un composant clé du système de supervision :

- **Nagios Core**, en tant que moteur principal de supervision, permet le suivi des services et hôtes sur le réseau.
- **NCPA** est utilisé comme agent de supervision installé sur les machines clientes, permettant une collecte précise des métriques système.
- **NagiosQL**, un outil de gestion graphique, facilite la création et la modification des fichiers de configuration.
- Enfin, **Thruk**, une interface Web moderne, vient enrichir l'affichage et l'analyse des données collectées.

Ce regroupement méthodique a pour but non seulement de justifier les choix techniques effectués, mais également de servir de guide pratique de référence, aussi bien pour une réinstallation future que pour d'autres projets similaires.

Annexe 1 : Installation et Configuration de Nagios

Objectif

Cette annexe décrit l'ensemble des étapes nécessaires à l'installation et à la configuration de **Nagios Core**, moteur principal de supervision, sur une machine Debian/Ubuntu. L'objectif est de mettre en place une plateforme fonctionnelle permettant de superviser des services, hôtes et ressources réseau.

Pré-requis

- Système : Debian 11 ou Ubuntu 20.04
- Accès root (sudo)
- Connexion Internet active
- Paquets de développement et serveur web Apache

Étapes d'installation

1. Installation des paquets nécessaires :

```
sudo apt install wget unzip curl openssl build-essential libgd-dev libssl-dev \
libapache2-mod-php php-gd php apache2 -y
```

2. Téléchargement des sources Nagios :

```
wget https://assets.nagios.com/downloads/nagioscore/releases/nagios-4.5.9.tar.gz
```

3. Extraction de l'archive

```
sudo tar -zxvf nagios-4.5.9.tar.gz
cd nagios-4.5.9
```

4. Configuration et compilation de Nagios

```
sudo ./configure
sudo make all
```

5. Création des groupes et utilisateurs

```
sudo make install-groups-users
sudo usermod -a -G nagios www-data
```

6. Installation de Nagios

```
sudo make install  
sudo make install-init  
sudo make install-commandmode  
sudo make install-config  
sudo make install-webconf
```

7. Configuration d'Apache

- **Activation des modules :**

```
sudo a2enmod rewrite  
sudo a2enmod cgi
```

- **Redémarrage du service :**

```
sudo systemctl restart apache2
```

8. Création d'un utilisateur pour l'interface Web

```
sudo htpasswd -c /usr/local/nagios/etc/htpasswd.users nom_utilisateur
```

Installation des plugins Nagios

1. Téléchargement :

```
cd ~/  
wget https://nagios-plugins.org/download/nagios-plugins-2.3.3.tar.gz
```

2. Extraction et préparation :

```
sudo tar -zxvf nagios-plugins-2.3.3.tar.gz  
cd nagios-plugins-2.3.3/  
sudo ./configure --with-nagios-user=nagios --with-nagios-group=nagios
```

3. Compilation et installation :

```
sudo make  
sudo make install
```

Vérification et démarrage de Nagios

1. Vérifier la configuration :

```
sudo /usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

2. Démarrer Nagios et activer au démarrage :

```
sudo systemctl start nagios  
sudo systemctl enable nagios
```

3. Accès à l'interface Web :

<https://localhost/nagios/>

NB : Se connecter avec l'utilisateur nagiosadmin (ou autre) créé précédemment.

Annexe 2 : Installation et Configuration de NCPA

Objectif

Cette annexe décrit les étapes d'installation et de configuration de l'agent NCPA sur une machine cliente. NCPA permet à Nagios Core de superviser les ressources système (CPU, mémoire, disque, etc.) via une API sécurisée. L'objectif est d'intégrer cette machine à la supervision centrale.

Pré-requis

- Serveur Nagios opérationnel
- Accès root sur la machine cliente
- Connexion réseau entre le serveur et le client
- Port TCP 5693 ouvert

Télécharger NCPA pour Windows 11

1. Télécharger NCPA (lien) :

<https://www.nagios.org/projects/ncpa/#download-ncpa-section>

2. Choisir la version Windows :

Sur la page de téléchargement, clique sur le lien pour Windows. Cela te redirigera vers un fichier d'installation .exe pour Windows.

3. Télécharger le fichier d'installation :

Télécharge le fichier NCPA Installer pour Windows (un fichier .exe).

Installer NCPA sur Windows 11 Guide complet

<https://assets.nagios.com/downloads/nagiosxi/docs/Installing-NCPA.pdf>

NB : choisir un token sécuriser

Vérifier que NCPA fonctionne sur Windows 11

1. Accéder à l'interface Web de NCPA :

Ouvre ton navigateur et tape l'URL suivante pour accéder à l'interface de NCPA :

`https://:5693` Remplace par l'adresse IP locale de ton PC Windows 11.

Par exemple, si ton adresse IP locale est 192.168.1.100 , entre `https://192.168.1.100:5693` .

2. Se connecter à l'interface :

Tu seras invité à entrer le mot de passe que tu as défini pendant l'installation pour accéder à l'interface Web de NCPA. Une fois connecté, tu devrais voir des informations sur ton système Windows, comme l'utilisation du processeur, de la mémoire, du disque, etc.

Configurer Nagios Core pour surveiller NCPA

Maintenant que NCPA est installé sur ta machine Windows, tu dois configurer Nagios Core pour qu'il puisse surveiller cette machine.

1. Télécharger et installer le plugin `check_ncpa` sur le serveur Nagios

`curl -O`

`https://raw.githubusercontent.com/NagiosEnterprises/ncpa/master/agent/plugins/check_ncpa.py`

- **Placer le plugin sur ton serveur Nagios :**

Une fois téléchargé, place le fichier du plugin `check_ncpa` dans le répertoire des plugins Nagios.

Sur une installation par défaut, ce répertoire est souvent situé dans `/usr/local/nagios/libexec/` .

`sudo cp check_ncpa.py /usr/local/nagios/libexec/`

- **Rendre le plugin exécutable :**

Sur ton serveur, donne les droits d'exécution au plugin :

`sudo chmod +x /usr/local/nagios/libexec/check_ncpa.py`

2. Ajouter un hôte dans Nagios pour surveiller Windows 11

- **Modifier le fichier de configuration des hôtes :**

Sur ton serveur Nagios, va dans le répertoire où sont stockées les configurations des hôtes, souvent dans `/usr/local/nagios/etc/objects/` .

Crée un fichier pour ton hôte Windows 11 ou modifie un fichier existant.

- **Ajouter un nouvel hôte :**

```
sudo nano /usr/local/nagios/etc/objects/windows.cfg
```

- **Ajoute une entrée pour ton hôte Windows 11.**

Voici un exemple de configuration d'hôte :

```
define host {  
    use                windows-server  
    host_name          Mon_PC_Windows  
    alias              PC de test Windows  
    address            192.168.1.25  
}
```

NB : Remplace l'IP par l'adresse de ton PC Windows.

3. Ajouter ce fichier dans nagios.cfg

```
sudo nano /usr/local/nagios/etc/nagios.cfg
```

```
cfg_file=/usr/local/nagios/etc/objects/windows.cfg
```

4. Ajouter une commande dans Nagios

```
sudo nano /usr/local/nagios/etc/objects/commands.cfg
```

```
define command {  
    command_name    check_ncpa  
    command_line    $USER1$/check_ncpa.py -H $HOSTADDRESS$ -t $ARG1$ -P 5693 -M $ARG2$ -w $ARG3$ -c  
$ARG4$  
}
```

5. Redémarrer Nagios :

Après avoir ajouté les hôtes et les services, tu dois redémarrer Nagios pour prendre en compte les modifications :

```
sudo systemctl restart nagios
```

Vérifier la surveillance dans Nagios

1. Accéder à l'interface Web de Nagios :

Ouvre un navigateur et accède à l'interface Web de Nagios Core, généralement accessible à cette adresse : <https://localhost/nagios/>

2. Vérifier que l'hôte Windows 11 est bien surveillé :

Sur l'interface, tu devrais voir l'hôte Windows 11 Host et les services associés (comme l'utilisation du CPU). Si tout est configuré correctement, Nagios affichera l'état de ces services.

Annexe 3 : Installation et Configuration de NagiosQL

Objectif

Cette annexe décrit l'installation complète de **NagiosQL 3.5.0**, une interface web permettant de gérer visuellement les fichiers de configuration de **Nagios Core** (hôtes, services, commandes, etc.) via un navigateur web. Cette méthode facilite la maintenance et l'administration du système de supervision.

Pré-requis système

- Système Ubuntu 20.04 ou 22.04
- Apache2, MariaDB, PHP 7.4 (⚠ PHP 8.x peut causer des erreurs)
- Nagios Core installé et fonctionnel
- Accès administrateur (sudo)

Étapes d'installation par commande

1. Installation des dépendances :

```
sudo apt update
sudo apt install apache2 mariadb-server php libapache2-mod-php php-mysql \
php-xml php-cli php-cgi php-common php-intl php-pear unzip wget -y
```

2. Installation spécifique de PHP 7.4 :

```
sudo add-apt-repository ppa:ondrej/php
sudo apt update
sudo apt install php7.4 libapache2-mod-php7.4 php7.4-mysql php7.4-gettext \
php7.4-xml php7.4-cli php7.4-common php7.4-cgi -y
sudo a2dismod php8.3 && sudo a2enmod php7.4
sudo systemctl restart apache2
```

3. Téléchargement et installation de NagiosQL :

```
cd ~/Downloads
wget https://master.dl.sourceforge.net/project/nagiosql/nagiosql/nagiosql-3.5.0-git2023-06-18.tar.gz
tar -xvzf nagiosql-3.5.0-git2023-06-18.tar.gz
sudo mv nagiosql-3.5.0 /var/www/html/nagiosql
```

```
sudo chown -R www-data:www-data /var/www/html/nagiosql
sudo chmod -R 755 /var/www/html/nagiosql
```

4. Configuration de la base de données :

- **Connexion à MariaDB :**

```
sudo mysql -u root -p
```

- **Création de la base et des utilisateurs :**

```
CREATE DATABASE nagiosql;
CREATE USER 'nagiosqluser'@'localhost' IDENTIFIED BY 'motdepassefort';
GRANT ALL PRIVILEGES ON nagiosql.* TO 'nagiosqluser'@'localhost';
CREATE USER 'nagiosqladmin'@'localhost' IDENTIFIED BY 'adminpass';
GRANT ALL PRIVILEGES ON nagiosql.* TO 'nagiosqladmin'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

5. Configuration PHP :

- **Éditer le fichier de configuration :**

```
sudo nano /etc/php/7.4/apache2/php.ini
```

- **Modifier ou ajouter les lignes :**

```
date.timezone = Africa/Casablanca
display_errors = On
error_reporting = E_ALL
```

- **Puis redémarrer Apache :**

```
sudo systemctl restart apache2
```

6. Correction de bug Access denied - CREATE USER :

- **Éditer le fichier de NagInstallClass :**

```
sudo nano /var/www/html/nagiosql/install/functions/NagInstallClass.php
```

- **Commenter cette partie dans le bloc `/* Grant NagiosQL database user */`**

```

/*if (($this->arrSession['install']['dbtype'] === 'mysql') ||
($this->arrSession['install']['dbtype'] === 'mysqli')) {
/* Does the NagiosQL database user exist?*/
$intUserError = 0;
$this->myDBClass->insertData('FLUSH PRIVILEGES');
$strSQL = "SELECT * FROM `mysql`.`user` WHERE `Host`='" . $this->arrSession['install']['localsrv'] .
        . "AND `User`='" . $this->arrSession['install']['dbuser'] . "'";
$intCount = $this->myDBClass->countRows($strSQL);
if ($intCount === 0) {
    $strSQL = "CREATE USER '" . $this->arrSession['install']['dbuser'] . "'@" .
        . $this->arrSession['install']['localsrv'] . "' "
        . "IDENTIFIED BY '" . $this->arrSession['install']['dbpass'] . "'";
    $booReturn = $this->myDBClass->insertData($strSQL);
    if ($booReturn === false) {
        $intUserError = 1;
        $strDBError = str_replace(':', '<br>', $this->myDBClass->strErrorMessage);
    }
}
}

```

Figure 0.1: Exemple code 1 NagInstallClass.php

jusqu'à la ligne :

```
$this->myDBClass->insertData('FLUSH PRIVILEGES');} }
```

- Ajouter ce bloc après la partie commenter :

```

if ($intUserError !== 1) {
    if ($intUserError === 2) {
        $strStatusMessage = '<span class="green">' . $this->translate('done') . '</span> (' .
            $this->translate('Only added rights to existing user') . ': ' .
            $this->arrSession['install']['dbuser'] . ')';
    } else {
        $strStatusMessage = '<span class="green">' . $this->translate('done') . '</span>';
    }
} else {
    $strErrorMessage .= $strDBError . "<br>\n";
    $strStatusMessage = '<span class="red">' . $this->translate('failed') . '</span>';
    $intReturn = 1;
}
return $intReturn;
}

```

Figure 0.2: Exemple code 2 NagInstallClass.php

- Redémarrer Apache :

```
sudo systemctl restart apache2
```

Etape d'installation via interface web

7. Aller dans un navigateur :

<http://localhost/nagiosql/install/index.php>

8. Remplir ces informations :

- **DB User** : nagiosqluser
- **DB Password** : motdepassefort
- **Admin DB User** : nagiosqladmin
- **Admin Password** : adminpass
- **Nagios config path** : /usr/local/nagios/etc
- **NagiosQL config path** : /usr/local/nagios/etc/nagiosql

9. Connexion à l'interface NagiosQL:

<http://localhost/nagiosql>

NB : Se connecter avec l'utilisateur nagiosqluser (ou autre) créé précédemment.

Configuration de NagiosQL

10. Accéder à NagiosQL :

<http://localhost/nagiosql/admin.php>

11. Importer les fichiers de configuration :

- **Aller dans Tools > Data Import**



Figure 0.3: NagiosQL data import

- **Importer les objets suivants un par un**
 - ✓ Hosts
 - ✓ Services
 - ✓ Contacts
 - ✓ Commands
 - ✓ Timeperiods

- Cliquer sur **Import** pour chaque type de configuration

12. Corriger les erreurs de type "file missing" :

Dans Supervision > Hosts ou Services, certaines lignes peuvent afficher “missing” dans la colonne File. Cela signifie que NagiosQL n’a pas encore généré les fichiers nécessaires, suivez ces étapes pour corriger ce problème :

- **Créer les dossiers requis :**

```
sudo mkdir -p /usr/local/nagios/etc/nagiosql/hosts
sudo mkdir -p /usr/local/nagios/etc/nagiosql/services
```

- **Créer les fichiers vides si besoin :**

```
sudo touch /usr/local/nagios/etc/nagiosql/commands.cfg
sudo touch /usr/local/nagios/etc/nagiosql/contacts.cfg
```

- **Appliquer les bonnes permissions :**

```
sudo chown -R www-data:www-data /usr/local/nagios/etc/nagiosql
sudo chmod -R 755 /usr/local/nagios/etc/nagiosql
```

13. Générer les fichiers de configuration :

- Cliquer sur **Write all config files**
- Un message vert devrait apparaître : *"Configuration files successfully written"*

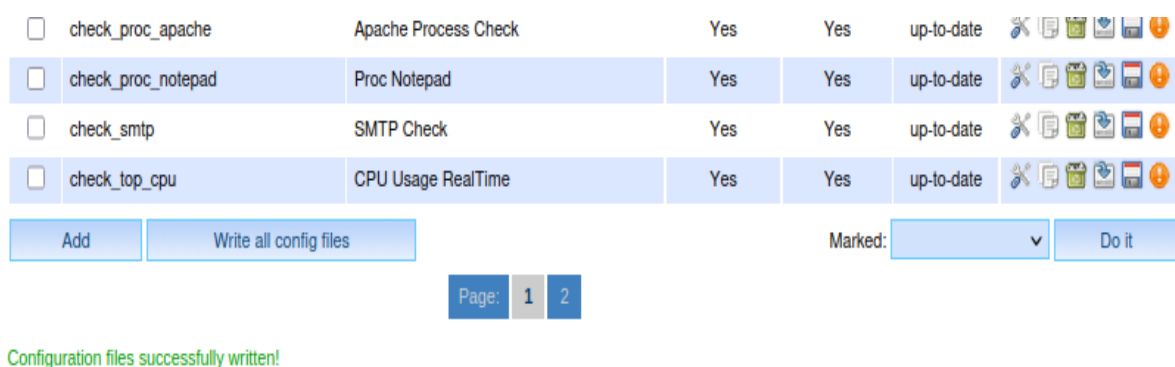


Figure 0.4: NagiosQL write all config

14. Intégrer les fichiers avec Nagios Core :

- **Accéder à :**

Administration > Config targets > [Modifier localhost]

Administration -> Administration -> Config targets

Main page
Supervision
Alarming
Commands
Specialties
Tools
Administration
New password
User admin
Group admin
Menu access
Data domains
Config targets

Configuration domain administration

Configuration target * localhost ?

Description * Local installation

Server name * localhost

Method Fileaccess v

Configuration directories

Base directory * /usr/local/nagios/etc/nagiosql/ ?

Host directory * /usr/local/nagios/etc/nagiosql/hosts/ ?

Service directory * /usr/local/nagios/etc/nagiosql/services/ ?

Backup directory * /usr/local/nagios/etc/nagiosql/backup/ ?

Figure 0.5: NagiosQL Config targets

- Mettre à jour les chemins :

Élément	Chemin
Base directory	/usr/local/nagios/etc/
Host directory	/usr/local/nagios/etc/hosts/
Service directory	/usr/local/nagios/etc/services/
Backup directory	/usr/local/nagios/etc/backup/
Host backup directory	/usr/local/nagios/etc/backup/hosts/
Service backup directory	/usr/local/nagios/etc/backup/services/
Nagios base directory	/usr/local/nagios/etc/
Import directory	/usr/local/nagios/etc/objects/
Nagios command file	/var/nagios/rw/nagios.cmd
Nagios binary file	/opt/nagios/bin/nagios
Nagios process file	/var/nagios/nagios.lock
Nagios config file	/usr/local/nagios/etc/nagios.cfg
Nagios CGI file	/usr/local/nagios/etc/cgi.cfg
Nagios resource file	/usr/local/nagios/etc/resource.cfg

Tableau 0.1: NagiosQL chemin modification

- Créer les répertoires manquants :

```
sudo mkdir /usr/local/nagios/etc/hosts/
sudo mkdir /usr/local/nagios/etc/services/
sudo mkdir /usr/local/nagios/etc/backup/
```

```
sudo mkdir /usr/local/nagios/etc/backup/hosts/  
sudo mkdir /usr/local/nagios/etc/backup/services/
```

- **Redémarrer le service Nagios :**

```
sudo systemctl restart nagios
```

15. Vérification dans l'interface Nagios Core :

- **Ouvrir :**

<http://localhost/nagios>

- **Aller dans les menus Hosts et Services :**

Vérifier l'apparition des objets configurés via NagiosQL.

Annexe 4 : Installation et Configuration de Thruk

Objectif

Cette annexe présente les étapes nécessaires pour installer et configurer **Thruk**, une interface web moderne permettant de visualiser les données de supervision provenant de Nagios via le module **Livestatus**. Thruk offre une alternative plus ergonomique à l'interface web native de Nagios.

Pré-requis

- Serveur Nagios déjà opérationnel
- Apache2 et PHP installés
- Accès root (sudo)
- Connexion Internet active

Étapes d'installation

1. Mise à jour du système

```
sudo apt update
```

```
sudo apt upgrade
```

2. Installation de Thruk et Apache/PHP

```
sudo apt install apache2 php
```

```
sudo apt install thruk
```

3. Configuration d'Apache pour Thruk

- **Modifier le fichier de configuration :**

```
sudo nano /etc/apache2/conf-enabled/thruk.conf
```

- **Y ajouter ou vérifier les lignes suivantes :**

```
ScriptAlias /thruk/cgi-bin/ "/usr/share/thruk/cgi-bin/"
```

```
Alias /thruk/ "/usr/share/thruk/html/"
```

```
Alias /thruk-static/ "/usr/share/thruk/static/"
```

```
<Directory "/usr/share/thruk">
  Options FollowSymLinks
  AllowOverride All
  Require all granted
</Directory>
```

- **Activer le module CGI et redémarrer Apache :**

```
sudo a2enmod cgi
```

```
sudo systemctl restart apache2
```

4. Accès à l'interface web Thruk

Dans le navigateur :

```
http://<adresse_IP_serveur>/thruk
```

5. Résolution de l'erreur 403 Forbidden (si nécessaire)

- **Éditer le fichier suivant :**

```
sudo nano /usr/share/thruk/thruk_cookie_auth.include
```

- **Remplacer les règles RewriteRule par celles-ci :**

```
RewriteRule ^/(.*)$
/auth:%1/%{REMOTE_ADDR}~~%{HTTP:Authorization}~~%{HTTP:X-Thruk-Auth-
Key}~~%{HTTP:X-Thruk-Auth-User}/____/$1/____/%{QUERY_STRING}
[C,NS,UnsafeAllow3F]

RewriteRule ^/(.*)$          ${thruk_users:$1/loginbad/} [C,NS,UnsafeAllow3F]
RewriteRule ^/pass/(.*)$     /$1 [NS,PT,L,E=!REMOTE_USER,UnsafeAllow3F]
RewriteRule ^/redirect/(.*)$ /$1 [NS,L,R=302,UnsafeAllow3F]
```

- **Création d'un utilisateur d'authentification**

```
sudo htpasswd /etc/thruk/htpasswd/ <nom_utilisateur>
```

Entrer un mot de passe sécurisé à l'invite.

6. Installation et configuration de Livestatus

```
cd /tmp
```

```
wget https://download.checkmk.com/checkmk/1.5.0p25/mk-livestatus-1.5.0p25.tar.gz
```

```
tar -xvzf mk-livestatus-1.5.0p25.tar.gz
```

```
cd mk-livestatus-1.5.0p25
```

- **Installer les dépendances :**

```
sudo apt install build-essential g++ make librrd-dev libboost-all-dev -y
```

- **Compiler Livestatus :**

```
./configure --with-nagios4
```

```
make
```

- **Créer le dossier et copier le module :**

```
sudo mkdir -p /usr/local/lib/mk-livestatus
```

```
sudo cp src/livestatus.o /usr/local/lib/mk-livestatus/livestatus.o
```

7. Intégration de Livestatus avec Nagios et Thruk

- **Modifier nagios.cfg :**

```
sudo nano /usr/local/nagios/etc/nagios.cfg
```

- **Ajouter :**

```
broker_module=/usr/local/lib/mk-livestatus/livestatus.o /usr/local/nagios/var/rw/live
```

- **Configurer Thruk pour pointer vers le socket :**

```
sudo nano /etc/thruk/thruk_local.conf
```

- **Ajouter :**

```
<Component Thruk::Backend>
```

```
  <peer>
```

```
    name = local
```

```
    type = livestatus
```

```
  <options>
```

```
    peer = /usr/local/nagios/var/rw/live
```

```
  </options>
```

```
  </peer>
```

```
</Component>
```

8. Redémarrage de Nagios et vérification

```
sudo systemctl restart nagios
```

```
sudo systemctl start apache2
```

9. Accès final

Interface Thruk accessible depuis :

<http://localhost/thruk>

Connexion avec les identifiants créés via htpasswd.

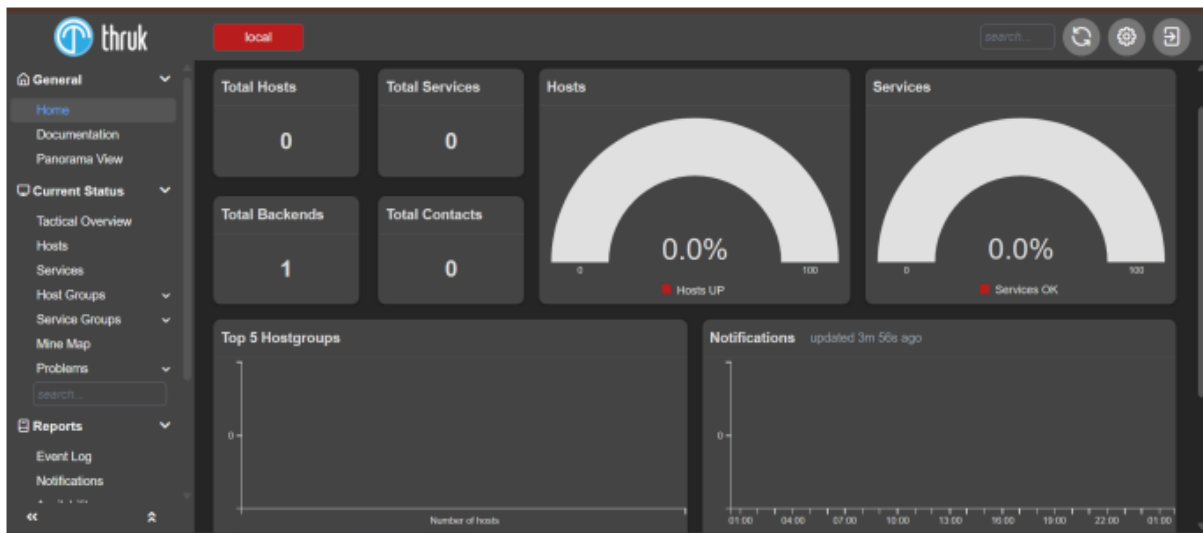


Figure 0.6: Interface thruk avant liaison

Configuration de Thruk

Compilation avancée de Livestatus (version 1.5.0p23, compatible Nagios 4.5.9)

Cette version permet une compatibilité parfaite avec Nagios 4.5.9, notamment au niveau des fichiers d'en-têtes requis pour la compilation.

Étapes détaillées :

1. Télécharger et compiler les sources de Nagios pour récupérer les headers nécessaires :

```
cd /tmp
```

```
export NAGVER=4.5.9
```

```
wget https://github.com/NagiosEnterprises/nagioscore/releases/download/nagios-  
${NAGVER}/nagios-${NAGVER}.tar.gz
```

```
tar -xzf nagios-${NAGVER}.tar.gz
```

```
cd nagios-${NAGVER}
```

```
./configure
```

```
make
```

```
make all
```

2. Créer les dossiers nécessaires à Thruk :

```
sudo mkdir -p /var/cache/thruk /var/lib/thruk
```

```
sudo chown -R www-data:www-data /var/cache/thruk /var/lib/thruk
```

3. Démarrer Apache (si ce n'est pas encore fait) :

```
sudo systemctl start apache2
```

4. Cloner le dépôt de mk_livestatus 1.5.0p23 :

```
cd /tmp
```

```
git clone https://github.com/Expensify/mk_livestatus mk_livestatus_150p23
```

```
cd mk_livestatus_150p23
```

5. Copier les headers de Nagios dans le dossier attendu par Livestatus :

```
rm -rf nagios4
```

```
cp -r /tmp/nagios-`${NAGVER}`/include nagios4
```

```
cp -r /tmp/nagios-`${NAGVER}`/lib lib
```

6. Compiler Livestatus :

```
autoreconf -f -i
```

```
./configure --prefix=/opt/nagios/mk-livestatus/1.5.0p23 --with-nagios4
```

```
make
```

```
sudo make install
```

7. Intégration finale dans Nagios

- Modifier le fichier nagios.cfg :**

```
sudo nano /usr/local/nagios/etc/nagios.cfg
```

- Ajouter ou modifier la ligne suivante :**

```
broker_module=/opt/nagios/mk-livestatus/1.5.0p23/lib/mk-livestatus/livestatus.o  
/usr/local/nagios/var/rw/live
```

8. Redémarrer Nagios :

`sudo systemctl restart nagios`

`sudo systemctl status nagios`

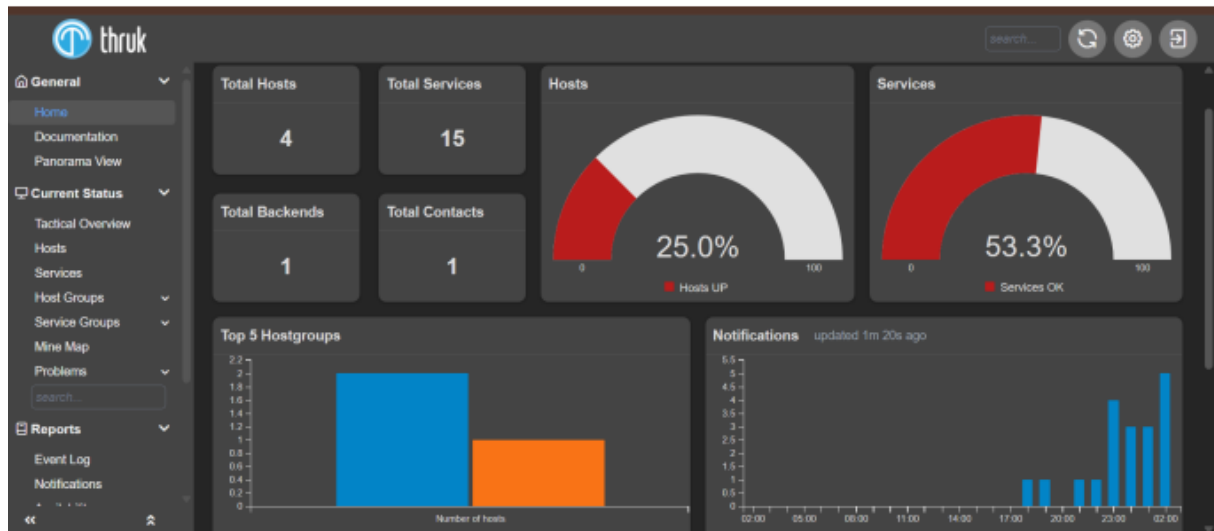


Figure 0.7: Interface thruk après liaison