

# MEMOIRE

## DE STAGE DE FIN D'ETUDES

Pour l'obtention du  
«Mastère professionnel en Nouvelles Technologies des  
Télécommunications et Réseaux (N2TR)»

Présenté par : Werghui Saida

Mise en place d'un outil de monitoring de réseau à base de logiciel libre

Soutenu le : 22 Juin 2019

Devant le jury :

**Président** : Mme Boucetta Hela

**Encadreur** : Mme Farhat Amel / Audouani Hedi

**Rapporteur** : Mme Lobna Kriaa

Année Universitaire : 2018 / 2019

# *Dédicace*

*On dédie ce travail à*

*Nos parents*

*Tous les membres de nos familles*

*Nos amis*

*Nos encadrateurs*

*Et tous ceux qui nous ont soutenus de près ou de loin  
tout au long de nos études*

*Merci à tous*

# *Remerciement*

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de l'Université Virtuelle de Tunis.

Avant d'entamer ce rapport, nous profitons de l'occasion pour remercier tout d'abord notre professeur Mme Farhat Amel n'a pas cessé de nous encourager pendant la durée du projet, ainsi pour son entière disponibilité et sa générosité en matière de formation et d'encadrement.

Nous tenons à remercier également nos encadreurs Mr Ben Wahida Slah, et Mr Hedi Audouani, pour leurs efforts de suivi pour l'aide précieuse qu'ils nous ont donnée et qui nous ont aidé à accomplir notre projet de fin d'étude dans des meilleures conditions.

Nos remerciements s'adressent au personnel du ministère des affaires sociales pour leur accueil chaleureux et leur esprit de collaboration si précieux. Et finalement, un grand merci aux membres du jury qui ont accepté d'évaluer notre travail.

## **RESUME**

Le présent rapport, effectué en ministère des affaires sociales de Tunis, est inscrit dans le cadre d'un mémoire de stage de fin d'étude pour l'obtention du «Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)» au niveau de l'Université Virtuelle de Tunis.

L'objet de ce projet est la Mise en place d'un outil de monitoring de réseau à base de logiciel libre nommé Shinken cela permet d'avoir une vue globales et en temps réel sur l'état du réseau informatique.

### **Mots clés**

Réseau, supervision, SMMP, Shinken

## **ABSTRACT**

This report makes the Ministry of Social Affairs of Tunis, is part of a dissertation of end of study for obtaining the "Professional Master in New Technologies of Telecommunications and Networks (N2TR)" at the Virtual University of Tunis.

The objective of this project is the establishment of a monitoring tool based on a free software called Shinken it helps to have a global view in real time accross the infrastructure

### **Key Words :**

Network, monitoring, SMMP, Shinken

## Cahier des charges

### Titre du projet

Mise en place d'un outil de monitoring de réseau à base de logiciel libre.

### Entreprise d'accueil

Ministère des affaires sociales

### Travail demandé

Recherche, Implémentation et configuration d'une solution open source, qui vise à superviser à distance les différents routeurs du ministère des affaires sociales avec gestion des alertes.

### Cadre du projet

Ce travail est inscrit dans le cadre d'un mémoire de stage de fin d'étude pour l'obtention du «Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)» au niveau de l'Université Virtuelle de Tunis

### Thème

Le but principal du projet est de pouvoir installer et mettre en place une station de surveillance des routeurs, pour les raisons suivantes :

- ✓ Coûts financiers les plus réduits possibles
- ✓ Récupération des informations permettant la détection des pannes
- ✓ L'indisponibilité des routeurs et de leurs services.
- ✓ Des renseignements supplémentaires de monitoring sur la charge CPU, Espace disque, mémoire disponible, etc...
- ✓ Gestion des alertes, avec notification par mail ou SMS en cas de problème
- ✓ Générer des rapports sur le fonctionnement des routeurs,...

## Sommaire

Dédicace.....	i
Remerciement .....	ii
RESUME et ABSTRACT.....	iii
Cahier des charges .....	iv
Sommaire.....	v
Table des figures .....	viii
Table des tableaux .....	viii
Introduction générale .....	ix
<b>Chapitre 1 : Entreprise d'accueil et cadre général du projet.....</b>	<b>1</b>
Introduction .....	1
I. Cadre général et présentation du projet .....	1
1. Cadre général du projet.....	1
2. Présentation du projet .....	1
II. Présentation de l'organisme d'accueil.....	1
1. Mission et attribution.....	1
2. Présentation de l'organigramme du ministère des affaires sociales.....	2
3. La direction de l'organisation, des méthodes et de l'informatique .....	3
III. Etude de l'existant.....	4
1. Description de l'existant.....	4
2. Critique de l'existant .....	7
3. Solution proposée .....	7
Conclusion.....	7
<b>Chapitre 2 : Supervision réseau et étude de choix .....</b>	<b>8</b>
Introduction .....	8
I. La supervision.....	8
1. Définition.....	8
2. Principe de la supervision.....	8
3. Intérêt de la supervision.....	8
II. Les protocoles de supervision réseau .....	9
1. Problématique.....	9
2. Exemple de protocoles de supervision.....	9
3. Composants et principe de fonctionnement du protocole SNMP .....	10
4. Les différents types d'actions du protocole SNMP.....	10

III. Etude et choix d'un outil de monitoring .....	11
1. Les solutions propriétaires .....	12
2. Les solutions libres .....	12
3. Tableau comparatif et choix de la solution .....	14
4. Choix de la solution .....	15
Conclusion.....	16
<b>Chapitre 3 : Etude de la solution shinken .....</b>	<b>17</b>
Introduction .....	17
I. La solution shinken.....	17
1. Les caractéristiques de shinken.....	17
2. Les fonctionnalités de shinken .....	18
II. Architecture de Shinken.....	19
1. Les composants du shinken.....	19
2. Architecture globale du Shinken .....	21
III. Extensions à shinken .....	21
1. Les greffons .....	21
2. Les modules.....	22
3. Plugins .....	22
4. Agents.....	22
IV. Principe de fonctionnement du système shinken.....	23
Conclusion.....	24
<b>Chapitre 4 : Spécification et conception des besoins .....</b>	<b>25</b>
Introduction .....	25
I. Spécification des besoins .....	25
1. Les besoins fonctionnels.....	25
2. Les besoins non fonctionnels .....	26
II. Le Langage de modélisation.....	27
1. Les vues statiques.....	27
2. Les vues dynamiques.....	27
III. Conception et modélisation des besoins .....	28
1. Les acteurs.....	28
Conclusion.....	34
<b>Chapitre 5 : Mise en place de la solution shinken .....</b>	<b>35</b>
Introduction .....	35
I. Architecture du travail .....	35

1. Environnement matériel .....	35
2. Environnement logiciel : .....	36
II. Mise en place de la solution.....	36
1. Préparation du serveur shinken .....	36
2. Installation de Shinken : .....	37
3. La configuration du serveur-shinken .....	41
4. Observation de l'état du serveur et des routeurs : .....	42
5. Notification .....	43
Conclusion.....	48
Conclusion Générale .....	49
Webographie.....	50
Annexe A .....	1
Annexe B .....	1
Annexe C.....	1
Annexe D .....	1



## Table des figures

Figure I- 1 : Organigramme du ministère .....	2
Figure I- 2 : architecture générale du réseau .....	4
Figure I- 3 : Maps de la Tunisie .....	5
Figure I- 4 : Maps de la Tunisie .....	5
Figure I- 5 : l'état Up du routeur de l'INTES.....	6
Figure I- 6 : Etat des routeurs .....	6
Figure II- 1 : Les ports UDP utilisés par le protocole SNMP.....	10
Figure II- 2 : Les échanges des requêtes SNMP.....	11
Figure III- 1: Les fonctionnalités de shinken via l'interface web intégrée.....	18
Figure III- 2: Les composants de Shinken.....	19
Figure III- 3 : Architecture globale shinken .....	21
Figure III- 4: Modules/Greffons utilisés par les daemons .....	23
Figure III- 5 : Principe de fonctionnement du système shinken .....	24
Figure IV- 1 : Diagramme de cas d'utilisation globale .....	29
Figure IV- 2: Diagramme de cas d'utilisation « Administrateur » .....	30
Figure IV- 3: Diagramme de cas d'utilisation « superviseur ».....	30
Figure IV- 4: Diagramme de cas d'utilisation « gestion des plateformes.....	31
Figure IV- 5: Diagramme de cas d'utilisation « gestion des équipements réseaux » .....	32
Figure IV- 6: diagramme de cas d'utilisation « gestion des utilisateurs ».....	33
Figure V- 1 : Architecture de travail .....	35
Figure V- 2 : Le répertoire /etc/shinken .....	38
Figure V- 3 : Interface de connexion.....	41
Figure V- 4 : Le fichier admin.cfg.....	41
Figure V- 5 : Etat du serveur shinken et des routeurs.....	42
Figure V- 6 : Etat du serveur shinken et des routeurs.....	42
Figure V- 7 : Etat des services.....	43

## Table des tableaux

Tableau II- 1 : Requêtes SNMP et ses actions .....	11
Tableau II- 2 : Tableau de spécification des paramètres .....	14
Tableau II- 3 : paramètres de choix en fonction des solutions open source.....	15
Tableau II- 4 : Somme des points pour chaque solution .....	15
Tableau IV- 1 : Description de cas d'utilisation : gestion de plateforme .....	31
Tableau IV- 2: Description de cas d'utilisation « gestion des équipements réseaux » .....	32
Tableau IV- 3 : Cas d'utilisation «gestion des utilisateurs».....	33

## Introduction générale

De nos jours, on parle de réseau local pour les moyennes entreprises et pour les grandes et importantes entreprises, on parle de réseaux de longues distances. Leurs parcs informatiques se composent de centaines d'équipements (ordinateurs, imprimantes, scanners, routeurs, switch, ...) interconnectés entre eux. Cette perspective, engendre un grand et majeur problème pour l'intervention et la maintenance d'une part, et la gestion des pannes d'autre part.

Le bon fonctionnement du réseau informatique, nécessite donc la mise en place d'un outil de supervision. Cette politique de sécurité permet de surveiller le réseau, de détecter les éventuelles intrusions et d'alerter le bon fonctionnement et le dysfonctionnement du système informatique.

Le dysfonctionnement d'un système, est déclaré par, des messages sur la console de supervision, l'envoi des courriers électroniques ou bien des SMS à l'administrateur réseau. Deux types d'outils de supervision d'un système informatique sont possibles : les outils de supervision réseau payantes et les outils de supervision réseau à base de logiciels libres. D'où l'intérêt de notre projet, dans le ministère des affaires sociales, notre société accueillante. Ce projet, a une mission d'installer et configurer un serveur de supervision réseau. Le serveur aide l'administrateur réseau, à surveiller son parc informatique, qui est étalé géographiquement sur une large zone.

Ce rapport, est structuré en cinq chapitres. Dans le chapitre 1, nous présenterons, en premier lieu, le cadre général du projet, avec une présentation de notre société accueillante. En second lieu, nous ferons une description et la critique de l'existant, du parc informatique. Dans le chapitre 2, nous nous occuperons, en premier lieu de la supervision et ses objectifs, avec une étude des protocoles de la supervision réseau, en second lieu, nous étudierons les solutions de supervision de réseau informatique, pour finaliser par un choix d'une solution adéquate. Via le chapitre 3, nous nous intéresserons, à l'étude de la solution choisie et son principe de fonctionnement. Le chapitre 4, est consacré à l'analyse et la spécification des besoins de la solution. Dans le chapitre 5, nous nous occuperons à l'implémentation et la mise en place et la de la solution.

Enfin, un bilan clôtura ce rapport et synthétisera l'expérience de notre projet. En annexe, des captures d'écran pour la solution existante, des logiciels utilisés, ainsi que l'installation de la solution.

# Chapitre 1 : Entreprise d'accueil et cadre général du projet

## Introduction

Pour le bon fonctionnement d'un réseau informatique, la mise en place d'un outil de supervision s'avère nécessaire. En effet, il permet de surveiller le réseau, de détecter les éventuelles intrusions, et d'alerter le dysfonctionnement du système informatique.

Dans ce chapitre, nous présenterons, d'abord, l'entreprise accueil à savoir le ministère des affaires sociales ainsi que ses domaines d'intervention. Ensuite, nous donnerons l'étude de l'existant concernant la supervision réseau du parc informatique du MAS. Enfin, nous poserons la problématique liée à l'outil utilisé et nous proposerons une solution de supervision plus adéquate.

## I. Cadre général et présentation du projet

### 1. Cadre général du projet

Ce travail est inscrit dans le cadre d'un mémoire de stage de fin d'étude pour l'obtention du «Mastère professionnel en Nouvelles Technologies des Télécommunications et Réseaux (N2TR)» au niveau de l'Université Virtuelle de Tunis.

### 2. Présentation du projet

Le projet consiste à concevoir et à mettre en place une solution de supervision réseau informatique à base de logiciel libre. Nous aurons donc une mission de chercher et étudier les outils de supervision d'un parc informatique. Ces outils, permettent la surveillance et l'analyse des différents comportements des équipements du réseau informatique. Le ministère des affaires sociales, est l'entreprise qui nous accueille pour réaliser notre projet.

Dans ce qui suit, nous présenterons l'entreprise d'accueil.

## II. Présentation de l'organisme d'accueil

### 1. Mission et attribution

La mission générale du Ministère des Affaires Sociales, consiste à mettre en œuvre la politique sociale de l'État. Cette politique vise à assurer un développement social équilibré entre les catégories et les générations composant la société et à consolider le bien-être social à travers la consécration des valeurs de l'auto-responsabilité dans :

- ✓ Les domaines du travail et des relations professionnelles,

- ✓ La santé et la sécurité du travail et la sécurité sociale,
- ✓ La promotion des catégories vulnérables et à besoins spécifiques,
- ✓ L'enseignement des adultes et le logement social,
- ✓ L'encadrement de la communauté tunisienne à l'étranger.

## 2. Présentation de l'organigramme du ministère des affaires sociales

La Figure I-1, représente l'organigramme du MAS.

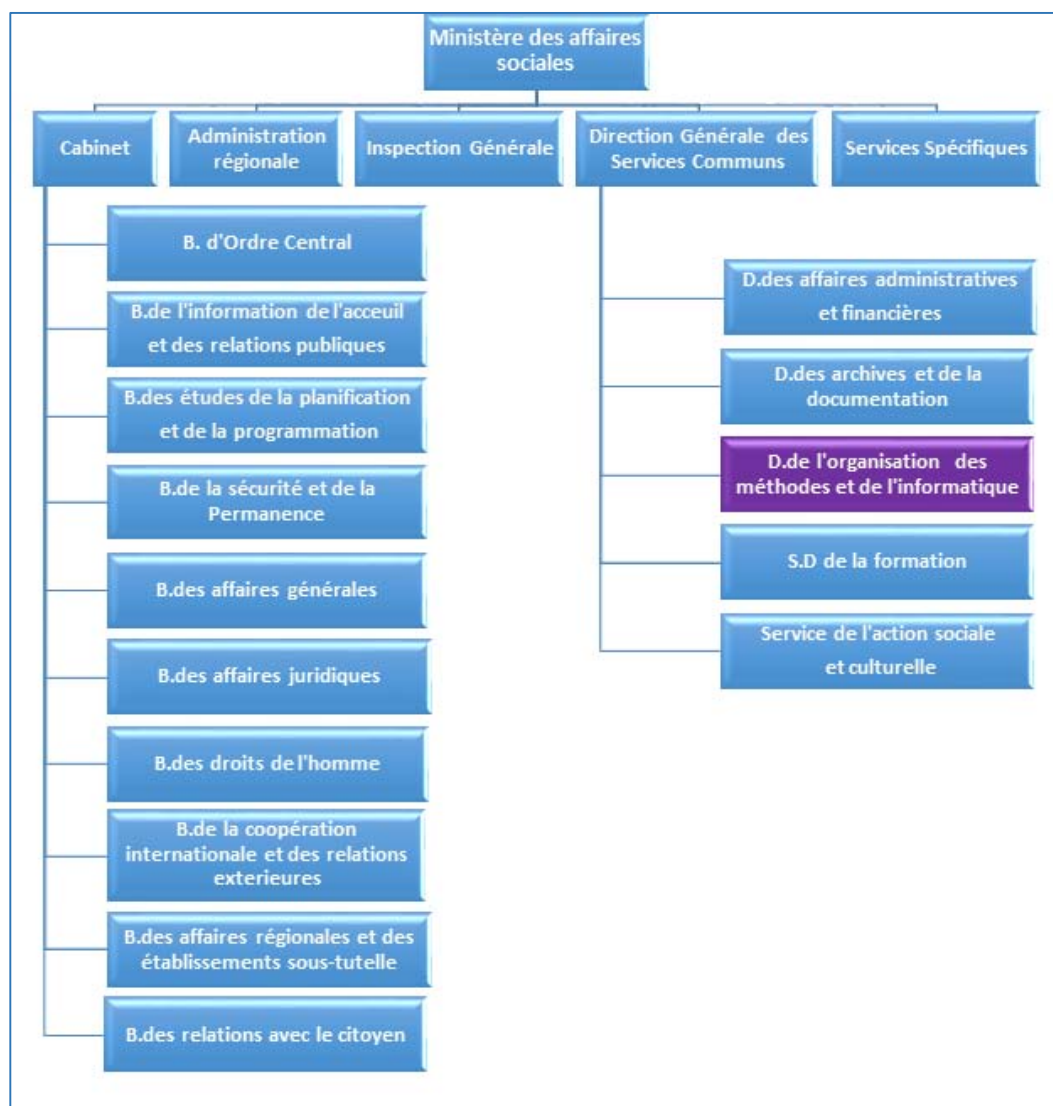


Figure I- 1 : Organigramme du ministère

Le ministère des affaires sociales comprend les directions suivantes :

Le Cabinet : dirigé par un chef de cabinet assisté par des chargés de mission ou des attachés de cabinet.

✓ Inspection générale : chargée du contrôle de la gestion administrative, technique et financière de l'ensemble des services de l'administration centrale du ministère, des services extérieurs et des établissements sous-tutelle, ainsi que des associations subventionnées par le budget du département.

✓ Les services spécifiques : comprennent La direction générale du travail, la direction générale de l'inspection du travail et de la conciliation, La direction générale de la sécurité sociale, ensuite la direction générale de la promotion sociale et en fin la direction de l'inspection médicale et de la sécurité au travail.

✓ L'administration régionale : Les attributions et l'organisation des directions régionales des affaires sociales sont fixées par décret.

La direction générale des services communs : comprend notamment plusieurs directions : la direction des affaires administratives et financières, la direction des archives et de la documentation, le service de l'action sociale et culturelle, et enfin la direction de l'organisation, des méthodes et de l'informatique. Notre projet sera effectué dans cette direction générale dans le service des projets de la sous-direction informatique. [\\_\(\[1\] s.d.\)](#)

### 3. La direction de l'organisation, des méthodes et de l'informatique

Cette direction est chargée notamment de coordonner l'activité du département en matière de réforme administrative, avec les services concernés du premier ministère, elle comprend deux sous-directions :

- ✓ La sous-direction de l'organisation et des méthodes
- ✓ La sous-direction de l'informatique

La sous-direction de l'informatique comprend trois services :

- Le service de l'exploitation
- Le service des études et de la formation
- Le service des projets

La sous-direction de l'informatique, a le but de développer l'utilisation de l'outil informatique au sein de l'administration par l'élaboration, la réalisation et le suivi du plan informatique du département. De plus, cette sous-direction a l'objectif d'assurer l'exploitation et la maintenance du matériel et du logiciel informatique.

Dans ce qui suit, nous présenterons, une étude de l'existant du parc informatique du ministère des affaires sociales.

### III. Etude de l'existant

#### 1. Description de l'existant

Le parc informatique du MAS, comporte des milliers d'équipements informatiques, tel que des machines (Windows et linux), des imprimantes, des scanners, des serveurs (DHCP, Web, Messagerie, et Base de données,...). De plus, le réseau du MAS, a une topologie en étoile, cette architecture présente plusieurs équipements réseaux, tous interconnectés au central du MAS. C'est une architecture flexible pour le dépannage et la gestion du réseau. La Figure I-2 représente l'architecture réseau du ministère.

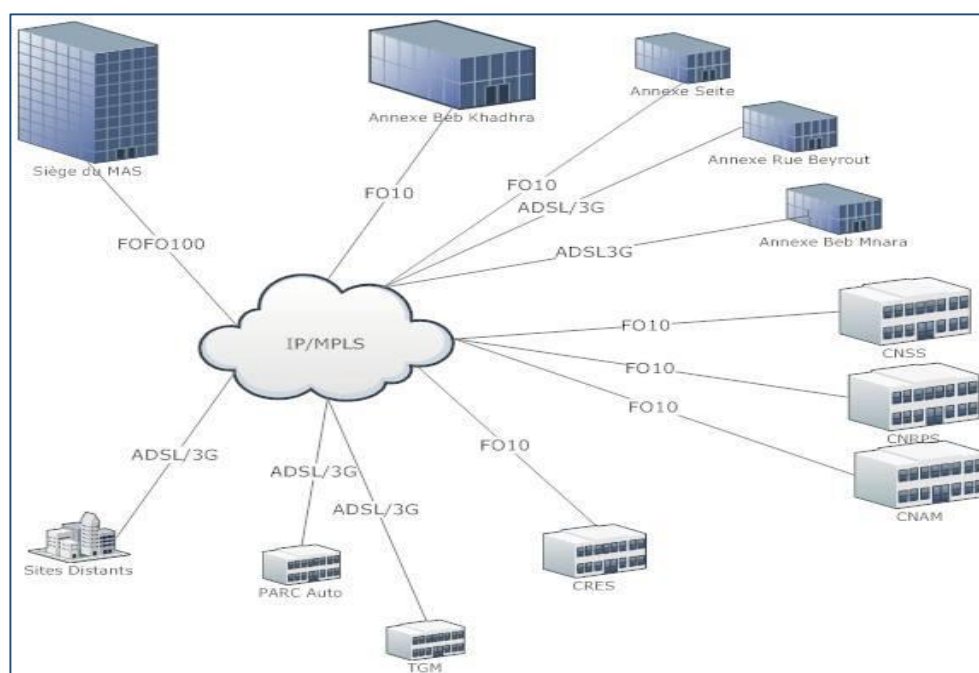


Figure I- 2 : architecture générale du réseau

Le ministère utilise une fibre optique de 100 Mbps dédoublée installée au niveau du siège, tous les sites sont connectés au VRF<sup>1</sup> (Virtual Routing and Forwarding) mis à la disposition du ministère au niveau du backbone<sup>2</sup> MPLS du fournisseur 3s/Global Net. Vu le grand nombre d'équipement (Routeurs, Serveurs, Switch,...), la supervision réseau du MAS est assurée par une solution payante nommée «SolarWinds». (Autres imprime-écrans dans «l' [Annexe A](#)»).

---

<sup>1</sup> **VRF** : est la virtualisation sur les routeurs : sur un routeur physique, on peut avoir plusieurs instances virtuelles et des tables de routage virtuelles qui sont séparées les unes aux autres.

<sup>2</sup> **Backbone** : Pour organiser efficacement les routeurs dans une grande zone (site), on partitionne une grosse base de données en zones plus petites et plus facile à gérer. La zone principale est la zone backbone, au quelle toutes les autres zones sont connectées directement.

La Figure I-3, représente le Maps de la Tunisie, qui montre que tous les gouvernorats sont interconnectés à la centrale.

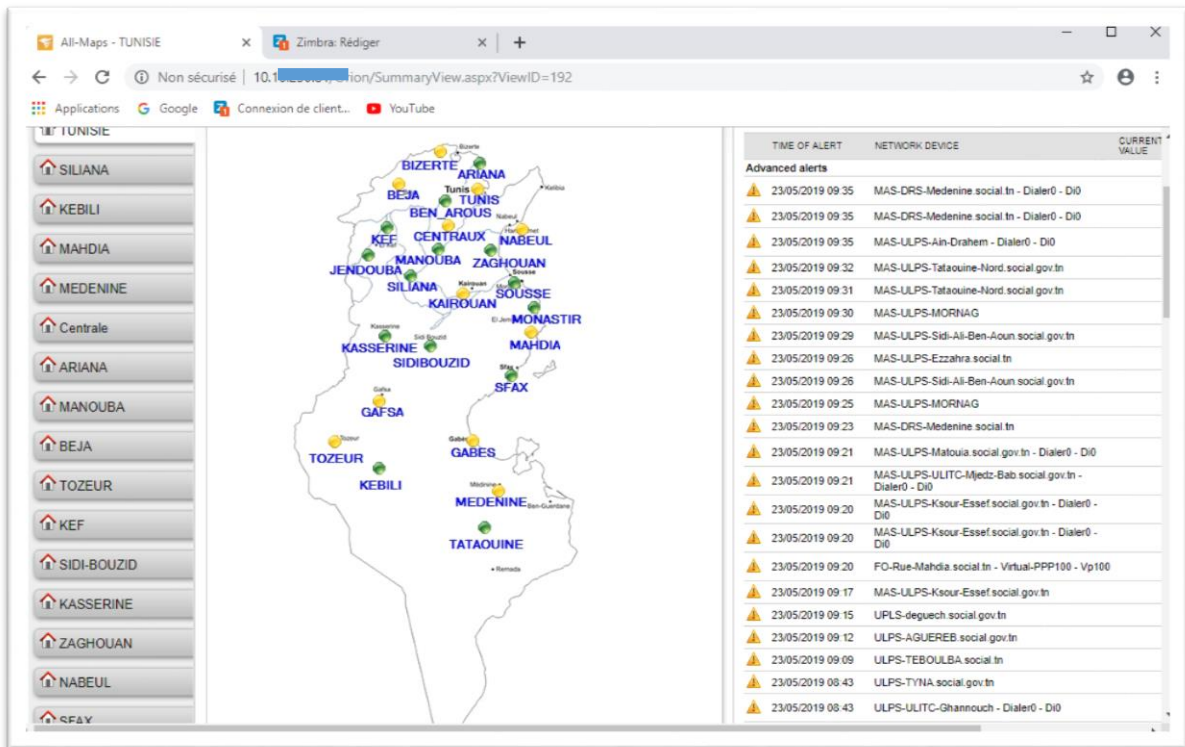


Figure I- 3 : Maps de la Tunisie

La Figure I-4, montre le Maps ainsi l'état de ses routeurs de la centrale.

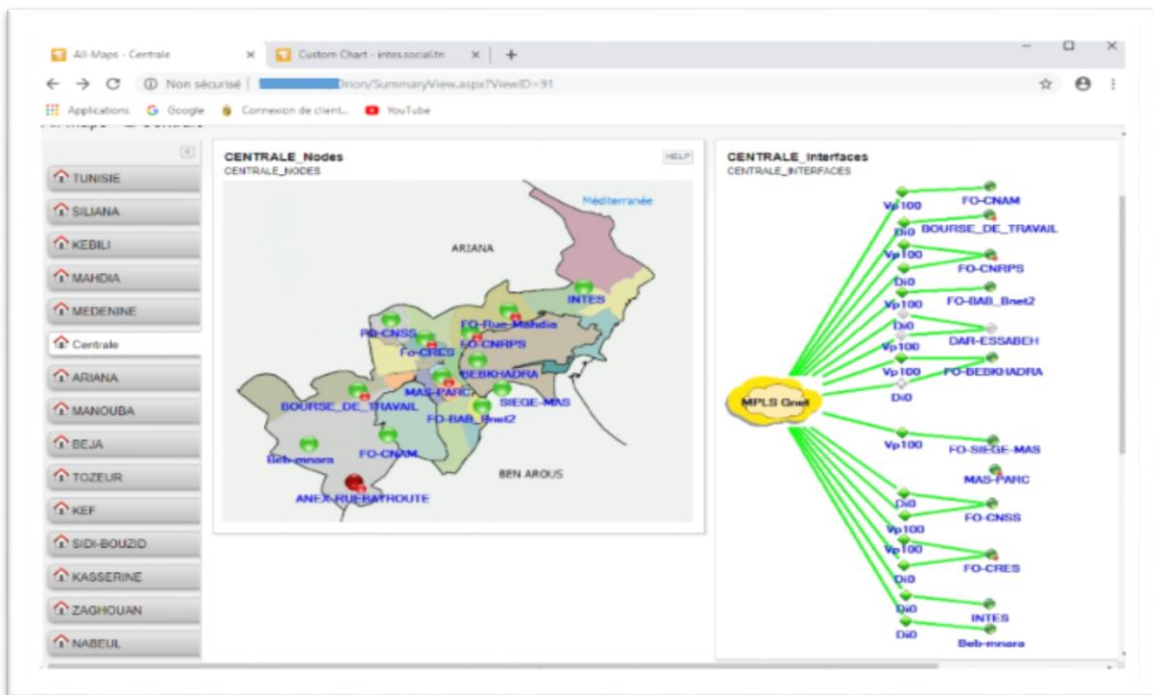


Figure I- 4 : Maps de la Tunisie



La Figure I-5, montre l'état Up du routeur de l'INTES (Institut National de Travail des Etudes Sociales).

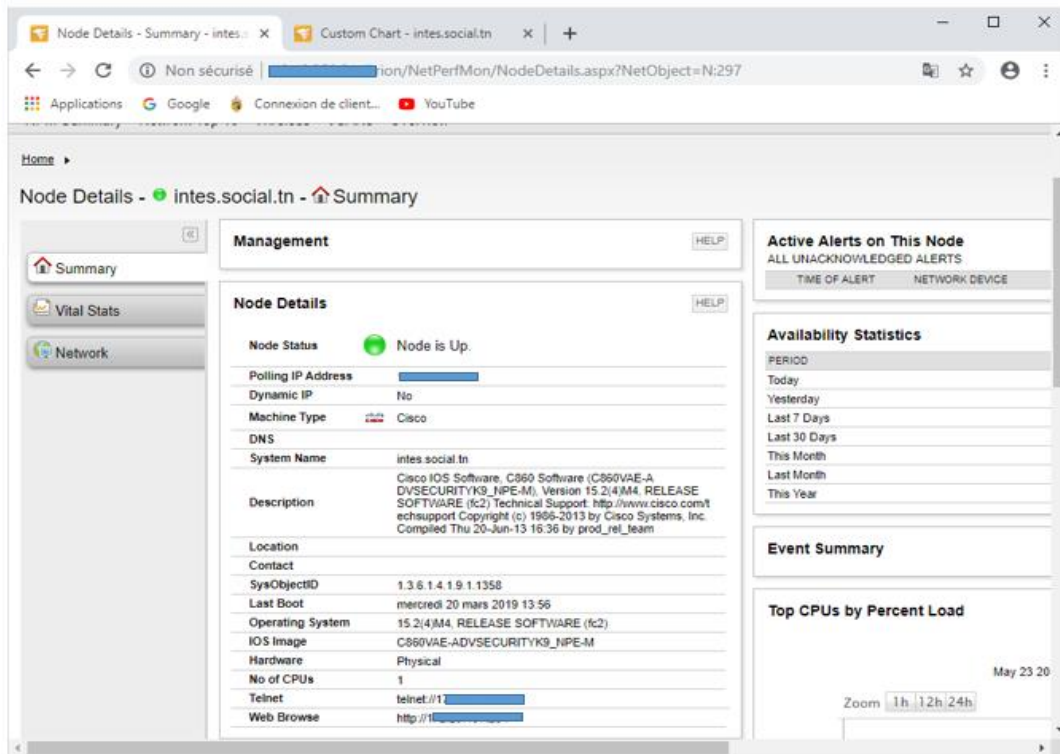


Figure I- 5 : l'état Up du routeur de l'INTES

La Figure I-6, montre l'état Up, Down, Warning et Unknown des routeurs de tous les gouvernorats.

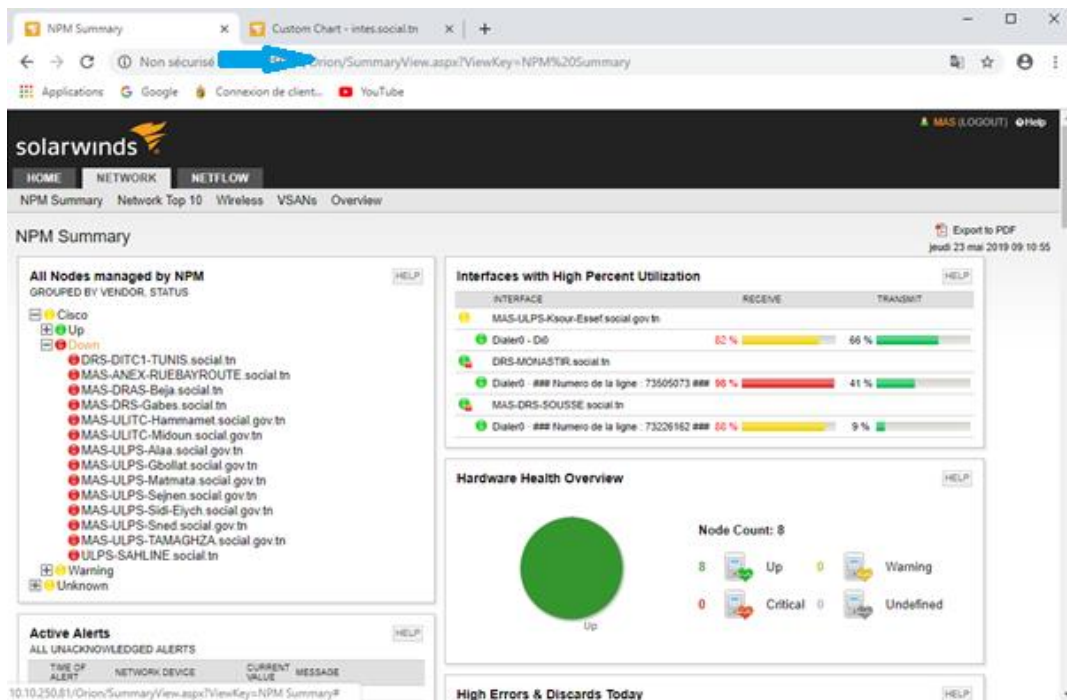


Figure I- 6 : Etat des routeurs



## 2. Critique de l'existant

Cet outil a plusieurs inconvénients, les jours de formation et les mises à jour de ce produit sont payants. SolarWinds n'est pas administrable et le code source n'est pas fourni. Ainsi que chaque module ajouté sera payé.

## 3. Solution proposée

Pour satisfaire le besoin de l'entreprise d'accueil, nous avons pensé à mettre en place un outil de monitoring de réseau à base de logiciel libre. Cette solution donnera la possibilité à l'administrateur d'intervenir plus rapidement en cas de pannes, de gérer son réseau de la manière voulue et d'offrir au MAS un gain financier important.

## Conclusion

Dans ce chapitre, nous avons présenté, d'abord, le ministère des affaires sociales, notre société d'accueil, ainsi que ses domaines d'intervention. Ensuite, nous avons présenté une étude de l'existant concernant la supervision réseau du parc informatique du MAS. Enfin, nous avons éclairé la problématique liée à l'outil utilisé et nous avons proposé une solution de supervision réseau plus adéquate.

Avant d'étudier les solutions de supervision, dans le chapitre suivant, nous définirons les notions de base de la supervision et ses protocoles.

## Chapitre 2 : Supervision réseau et étude de choix

### Introduction

Superviser un réseau, c'est choisir un bon outil de supervision, qui permet de surveiller, analyser, apporter et alerter les fonctionnements normaux et anormaux des systèmes informatiques.

Dans ce chapitre, nous présenterons, d'abord, la supervision avec son principe et ses intérêts. Ensuite, nous nous occuperons, des protocoles de supervision réseau. Puis, nous ferons une étude des solutions de supervision de réseau, pour finaliser par un choix d'une solution adéquate.

### I. La supervision

#### 1. Définition

En 1980, est apparu le concept de supervision, suite à la croissance importante de mise en place de réseau informatique dans les entreprises et leur administration. La supervision est une technique qui fait référence à la surveillance du bon fonctionnement d'un réseau, ainsi que les services informatiques connectés sur ce réseau en temps réel. Superviser c'est surveiller l'état du système, analyser et alerter les fonctionnements normaux et anormaux. [\\_\[2\] s.d.\)](#)

#### 2. Principe de la supervision

Pour une performante rentabilité, la supervision d'un parc informatique, vise plusieurs points et sensibilise plusieurs niveaux du parc. Principalement la supervision réseau informatique, s'occupe d'interconnexion et de la communication entre les équipements réseaux (routeur, switch, commutateur,...) qui sont indispensables pour la supervision. Deuxièmement, la supervision des systèmes informatique (système de plateforme de supervision : OS, capacité disque, mémoire, ...). En fin la supervision des applications qui permet de surveiller et agir sur ces dernières en temps réel. [\\_\(\[3\] s.d.\)](#)

#### 3. Intérêt de la supervision

La supervision, a pour objectif le contrôle de l'infrastructure du parc informatique (réseaux, systèmes, services applications,...). Il est nécessaire d'avoir une claire visibilité sur l'ensemble du parc à gérer et collecter les informations sur un tableau de bord. La supervision se base sur plusieurs points qui peuvent nous garantir ses services :

- ✓ Etre proactif : l'administrateur doit être actif pour agir rapidement et anticiper les pannes
- ✓ Etre réactif : prendre une décision rapide en mesurant les enjeux et limiter les risques en cas de problème.
- ✓ Cibler des pannes et les problèmes avant qu'ils apparaissent
- ✓ Intervention à distance, pour régler certains problèmes (redémarrage d'un serveur, d'un routeur, lancer des mises à jour, ...)
- ✓ Alerter l'administrateur réseau par un SMS ou par un E-mail, en cas de panne
- ✓ (etc...)

## II. Les protocoles de supervision réseau

### 1. Problématique

Comme nous avons dit précédemment, la base de la supervision, est les équipements d'interconnexion et de communication (Routeur, switch,...). Afin de contrôler et gérer le parc informatique, il faut interconnecter la totalité des hôtes. Le véritable obstacle dans ce cas, est comment faire communiquer et essentiellement contrôler efficacement en temps réel, les équipements à superviser ? C'est là où nous pouvons parler du protocole de supervision réseau.

### 2. Exemple de protocoles de supervision

Plusieurs outils et protocoles, permettent le contrôle et l'automatisation de la gestion d'équipements informatiques à distance, dans ce qui suit citerons quelques protocoles :

- ✓ IPMI : IPMI (Intelligent Platform Management Interface), c'est une interface intelligente intégré à certains matériel (serveurs) qui permet la gestion et le contrôle de certains paramètre à distance (comme température, sonde,...). [\\_\[4\] s.d.\)](#)

- ✓ Syslog : C'est un protocole qui permet la transmission d'évènements systèmes de chaque équipement et les centralisés dans une seule machine dans le but d'archivage, d'analyse et la production d'alerte. [\\_\[5\] s.d.\)](#)

- ✓ SNMP : SNMP (Simple Network Management Protocol), est un protocole utilisé par les administrateurs, pour gérer les équipements du réseau à distance. SNMP, est le protocole de communication le plus utilisé pour la gestion d'un réseau informatique. [\\_\[6\] s.d.\)](#)

Dans ce qui suit, nous étudierons le protocole SNMP.

### 3. Composants et principe de fonctionnement du protocole SNMP

La gestion d'un réseau informatique est composée de :

- ✓ Un superviseur (Manager de supervision)
- ✓ Des nœuds (Agent)
- ✓ Et une base de données (MIB)

Le protocole SNMP, fonctionne avec ces trois éléments, le superviseur, est une console de supervision, qui permet d'exécuter des requêtes de management et qui reste à l'écoute, des éventuels fonctionnements anormaux. Les nœuds (agents), sont les équipements réseaux administrables, que les administrateurs souhaitent superviser. Enfin, la base de données, est la MIB (Management Information Base), où sont enregistrées toutes les informations des évènements de la gestion. [\\_\[7\] s.d.\)](#)

La communication entre les équipements réseaux, se fait via le protocole SNMP, la transmission de données entre deux entités, dotée chacune par une adresse IP et un numéro de port, est réalisée par le protocole UDP. Chaque agent, a une MIB, qui contient toutes ses informations, il reste à l'écoute d'éventuelle action. Le superviseur est chargé, d'interroger les agents et de fournir à l'administrateur, de toutes les informations récupérées de chaque MIB des équipements du réseau. Suite, aux informations de sa MIB, le manager peut faire des interprétations de ces informations, pour faire des rapports d'incidents.

### 4. Les différents types d'actions du protocole SNMP

Le manager, peut communiquer et dialoguer avec les agents. La Figure II-1, montre que l'agent SNMP, utilise le port UDP 161 pour recevoir les messages, alors que, le superviseur SNMP, utilise le port UDP 162. [\\_\[8\] \)](#)

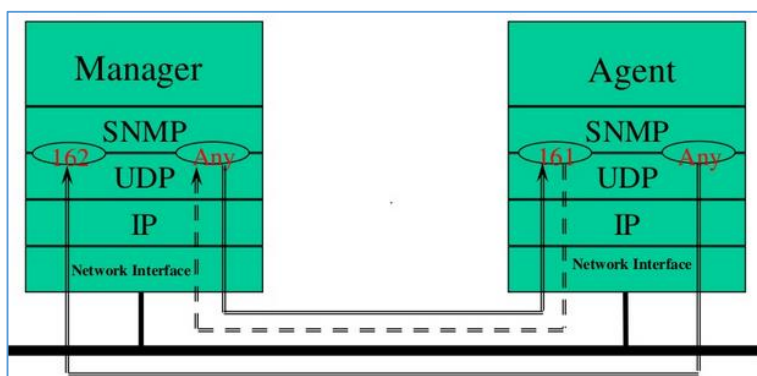


Figure II- 1 : Les ports UDP utilisés par le protocole SNMP

Ainsi, l'échange d'actions entre le manager et l'agent, est illustré par les requêtes SNMP et ses actions dans le Tableau II-1.

Tableau II- 1 : Requêtes SNMP et ses actions

Commande	Action
Get-request	Le superviseur SNMP demande une information à un agent SNMP
Get-next-request	Le superviseur SNMP demande l'information suivante à l'agent SNMP
Set-request	Le superviseur SNMP met à jour une information sur un agent SNMP
Get-reponse	L'agent SNMP répond à un get-request ou à un set-request
Trap	L'agent SNMP envoie une alarme au superviseur

La Figure II-2, résume les échanges des requêtes SNMP, entre le manager et son agent via les ports UDP 161 et 162. ([9] s.d.)

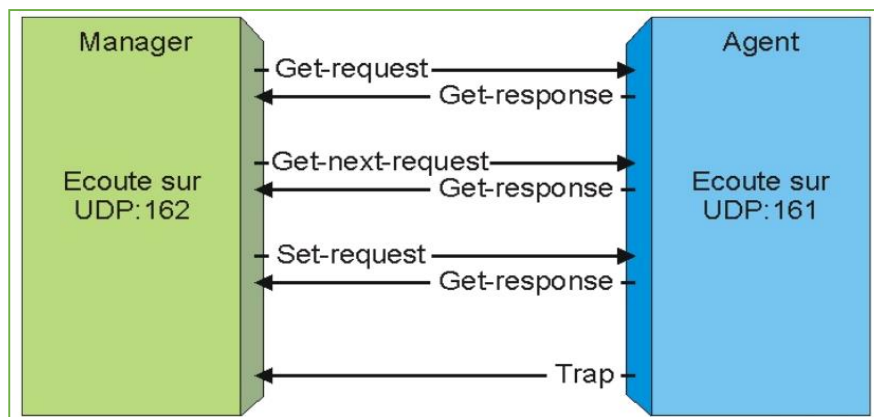


Figure II- 2 : Les échanges des requêtes SNMP

La supervision réseau, est assurée par le protocole SNMP. Nous détaillerons dans le volet suivant, les solutions de supervision réseau.

### III. Etude et choix d'un outil de monitoring

Dans le domaine informatique, plusieurs solutions de supervision disponibles, propriétaires ou open source. Une supervision efficace, doit obligatoirement réalisée par une des solutions de monitoring. Une étude de choix sur les solutions propriétaires et les solutions libres, sera faite dans la partie suivante.

## 1. Les solutions propriétaires

Les grandes sociétés ont compris l'importance de la supervision (systèmes, réseaux, services, sites web, trafics,...). Ces entreprises ont développé des outils de supervision pour avoir une vue globale de leur parc informatique et de son état. Il existe, plusieurs solutions (HP OpenView, WorksCisco, ITM,...). Dans ce qui suit, on se limitera à quelques exemples.

### a. HP OpenView



est un outil de supervision de la maison HP, caractérisé par une console offrant une vue globale du système d'information et une localisation des incidents. Le serveur contenant l'outil de monitoring, récolte les informations des états et des services de toutes les hôtes, dans une base de données. [\\_\(\[10\] s.d.\)](#)

### b. ITM (IBM Tivoli Monitoring)



Est un logiciel de supervision de l'entreprise IBM. Cet outil est assez complet en monitoring, mais plus puissant en métrologie que la supervision. Ce produit, fonctionne de manière proactive, pour surveiller les ressources, les systèmes et les composants vitaux de l'infrastructure, en détectant les problèmes potentiels. ITM, rend ses rapports de supervision, dans des tableaux et sous forme de graphes. [\\_\(\[10\] s.d.\)](#)

### c. PRTG

Est un logiciel de supervision des systèmes informatiques, en temps réel, de l'entreprise PAESSLER, il permet d'alerter des problèmes avant qu'ils ne se manifestent. [\\_\(\[11\] \)](#)

Les solutions propriétaires, exigent une mise en place complète du logiciel et du matériel. Le prix élevé, est le seul point commun de ces solutions. En outre, les critères des solutions propriétaires, rend ces derniers, non valables. En effet, aujourd'hui il est bien possible d'avoir un système d'information dans une entreprise, basé uniquement sur des logiciels libres.

## 2. Les solutions libres

Les solutions de monitoring open source, ne nécessitent pas d'être sous contrat d'infogérance avec le prestataire informatique. Ainsi, la gratuité et la disponibilité du code source, rend ces solutions très utilisées, dans plusieurs entreprises. Pour répondre à tous les


besoins de monitoring réseau, il existe, plusieurs outils de supervision open source comme Nagios, Zabbix, Centreon, Icinga, Cacti, Shinken,... Après, de multiples recherches, nous étudierons les quatre solutions les plus populaires.

#### a. Nagios

**Nagios**<sup>®</sup> : En 1999, Ethan Galstad a créé Nagios, le premier logiciel de supervision d'un système d'information open source. Il permet de visualiser et surveiller les équipements et leurs états, et de produire des rapports d'activité.

Cet outil, fonctionne sous la plupart des systèmes Unix et Linux. Ses plugins sont écrits en C. Nagios, est beaucoup plus attractif que ses produits concurrents, il récolte périodiquement et centralise les informations par un fonctionnement modulaire. Alors que sa configuration est complexe, elle nécessite une bonne connaissance de ce logiciel et il fonctionne uniquement sur une petite infrastructure non distribuée d'un millier d'équipements. [\\_\(\[12\] s.d.\)](#)


#### b. ZABBIX

 : Zabbix, est écrit en C par Alexei Vladishev, il fonctionne sur les systèmes, Linux, UNIX et Windows. Zabbix, se caractérise par la simplicité d'installation, d'utilisation, et d'apprentissage. Les données collectées, sont enregistrées dans une base de données de type MySQL, PostgreSQL ou Oracle. Le logiciel est composé d'un :

- ✓ un serveur, pour gérer le parc informatique, et envoie des alertes dans le cas de pannes.
- ✓ Un proxy, permet de protéger les données transitant sur le réseau.
- ✓ Des agents, récupèrent les données des machines supervisées et les renvoyées au serveur.

Zabbix, n'intègre pas certaines fonctionnalités, puisqu'il n'est pas possible d'ajouter de plugins. De même, l'interface graphique manque de lisibilité sur certains écrans de contrôle. [\\_\(\[13\] s.d.\)](#)

#### c. Shinken

 : La solution shinken, est un fork du projet nagios en Python, qui propose beaucoup de similitudes avec ce dernier. Tout d'abord, à l'image de nagios, les plugins respectant la syntaxe imposée sur les entrées sorties sont compatibles avec shinken, ce qui

rend l'application tout autant riche que nagios. Nagios fonctionne avec un seul daemon, alors que shinken à une architecture distribuée avec six daemons. [\\_\(\[14\] s.d.\)](#)

#### d. Centreon



Centreon est un logiciel libre de supervision, édité par la société française Merethis. Il permet de générer des rapports sur les incidents. Les données sont stockées dans une base de données MySQL. L'interface web, est multiutilisateur et gère des listes de contrôle d'accès, pouvant être paramétrées de manière très précise. Bien qu'il fonctionne par défaut avec nagios. [\\_\(\[15\] s.d.\)](#)

### 3. Tableau comparatif et choix de la solution

Dans le Tableau II-2, nous proposerons des paramètres de choix (OS, Interface web, performance,...) qui nous permettrons de faire le bon choix d'une solution, entre les solutions citées précédemment.

Tableau II- 2 : Tableau de spécification des paramètres

Base de Nagios	Interface web	Installation et configuration	Performance	Dépendance entre services	Protocole SNMP	Ecriture et ajout des plugins	Monitoring distribué	Utilisation des ressources	OS
1 =Non	1 = de base	5 = Très simple	1 = Faible	1 = Non	1 = Non	1 = Non	1 = Non	1 = Peu	1=Linux ou Windows
2 = Oui	2 = Moderne	4 = Facile	2 = Moyen	2 = Oui	2 = Oui	2 = Oui	2 = Oui	2 = Moyen	2= Linux et Windows
		3 = Moyen	3 = Bon					3 = Bon	
		2 = Compliqué	4 = Très bon					4 = Très bon	
		1 = Très difficile	5 = Excellent					5 =Excellent	



Le Tableau II-3, illustre, les paramètres de spécification en fonction des outils de supervision open source, décrites précédemment.

Tableau II- 3 : paramètres de choix en fonction des solutions open source

	Nagios	Zabbix	Centreon	Shinken
Base de Nagios	2	1	2	2
Performance	3	3	4	5
Installation et configuration	4	3	4	2
Utilisation des ressources	2	3	2	5
Dépendance entre services	2	1	2	2
Ecriture et ajout des plugins	2	2	2	2
Interface web ergonomique	1	1	2	2
Protocole SNMP	2	2	2	2
Langage de programmation	C	PHP/C++	PHP	Python
Monitoring distribué	2	1	2	2

En se basant, sur les spécifications des paramètres de choix, représentées dans le TableauII-2 et les paramètres de choix en fonction des solutions open source cités dans le Tableau II-3, le calcul de la somme des points pour chaque solution, nous donnera le résultat représenté dans le Tableau II-4 :

Tableau II- 4 : Somme des points pour chaque solution

Nagios	Zabbix	Centreon	Shinken
20	17	22	24

#### 4. Choix de la solution

Le tri des solutions, en se basant sur le Tableau II-4, nous a permis, de faire l'ordre du premier au dernier : shinken, Centreon, nagios et en fin Zabbix. Tout d'abord nous excluons Centreon, car il a beaucoup de versions ce qui provoque des problèmes de compatibilités, ainsi

il a certains modules qui sont payants. Nagios reconnu auprès des entreprises, mais, il maintienne à une partie payante, dérivante de sa partie libre, ce qui exige son élimination.

Après cette étude, nous semble que shinken est la solution adéquate à notre projet, au sein du ministère des affaires sociales.

- ✓ Shinken est totalement libre.
- ✓ Shinken a une interface plus moderne
- ✓ Shinken est un outil flexible, puissant, peut gérer un parc de plus de 200 serveurs grâce à son daemon qui peut être partagé en plusieurs parties contrairement aux autres outils

## Conclusion

Dans le chapitre 2, nous avons présenté, d'abord, la supervision et ses intérêts. Ensuite, nous avons étudié, les protocoles de supervision réseau. Puis, nous avons fait une étude des solutions de supervision réseau. En fin, et suite à notre étude comparative, nous avons opté, pour la solution open source shinken, qui sera étudiée de près dans le chapitre suivant.

## Chapitre 3 : Etude de la solution shinken

### Introduction

Dans ce chapitre, nous présenterons, d'abord, la solution shinken avec ses caractéristiques et ses fonctionnalités. Ensuite, nous ferons une étude de ses composants, en présentant son architecture globale. Puis, nous nous occuperons de ses extensions, pour finaliser par son principe de fonctionnement.

### I. La solution shinken

Shinken, est un logiciel de supervision réseau libre, sous licence GNU AGPL, créé en 2009 par Jean Gabès. Shinken, surveille les hôtes et les services spécifiés, en alertant les fonctionnements normaux et anormaux des systèmes. Shinken, a pour but d'apporter une supervision distribuée et hautement disponible facile à mettre en place. Il permet donc d'avoir une vision claire en temps réel, de l'état du réseau (système, services et applications) et aussi d'être prévenu directement en cas de défaillance (par mail, sms,...).[\(\[16\] s.d.\)](#)

#### 1. Les caractéristiques de shinken

Shinken, démarre comme une amélioration du projet nagios, il est caractérisé par :

- ✓ Une compatibilité totale avec nagios
- ✓ Une architecture modulaire, avec un code ouvert et réécrit en Python
- ✓ Un système distribué, fonctionnant sous Linux, Windows et Android
- ✓ Un système, avec une dépendance des services (découpage de rôle), et des sondes réutilisables
  - ✓ Nombreuses possibilités de développement des agents déployés sur les services distants (PHP, Perl, C, Python...)
  - ✓ La gestion du lien de parenté entre des éléments supervisés.
  - ✓ Un accès aux vues en fonction du profil utilisateur (intégration avec les annuaires standard)
  - ✓ Une interface web de visualisation intégrée WebUI
  - ✓ Une installation rapide avec documentation
  - ✓ ...

## 2. Les fonctionnalités de shinken

La solution shinken permet de collecter, toutes les informations du système informatique, pour une saine administration, du parc informatique. La Figure III-1\_([17] s.d.), montre les fonctionnalités, de la solution de supervision réseau shinken, qui permet :

- ✓ D'afficher la configuration, des ressources (CPU, RAM, DD, Carte Raid, Interface réseau,...)
- ✓ De visualiser en temps réel, les états courants, des équipements, des services réseaux et des applications
- ✓ D'afficher les historiques des évènements, dans des rapports détaillés
- ✓ D'afficher des graphiques, qui résument l'historique des données
- ✓ La génération des alertes, notification des administrateurs par envoi des Email ou SMS, suite à un mal fonctionnement du système
- ✓ Etc...

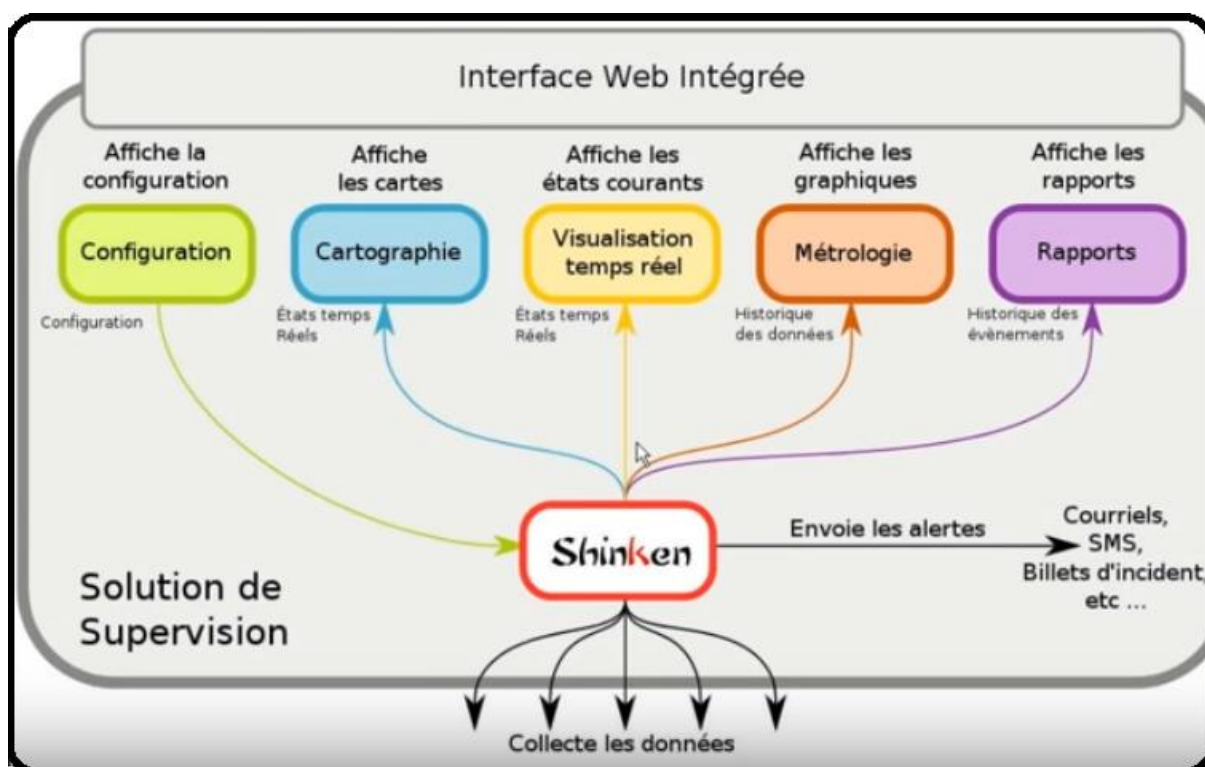


Figure III- 1: Les fonctionnalités de shinken via l'interface web intégrée

Shinken, est la solution, qui nous permet, d'avoir une vision claire en temps réel, de l'état de nos services, consulter les rapports, contenant les configurations, et l'historique des données et des évènements, ainsi d'être prévenu directement, par mail ou SMS en cas de mal fonctionnement du système.

## II. Architecture de Shinken

Shinken, est découpé en plusieurs parties, le cœur du système correspond aux six daemons, chaque démon à son rôle. Il y a également les modules, qui permettent d'ajouter des fonctionnalités au cœur de shinken et des plugins qui permettent d'effectuer les tests. Le point fort de shinken, c'est son architecture qui le différencie des autres outils de supervision.

### 1. Les composants du shinken

Regardons, les noms et les rôles des éléments, qui sont représentés dans la Figure III-2, nous pouvons remarquer, que chaque daemon, participe efficacement, dans la réalisation de la supervision. \_[18 ] s.d.)

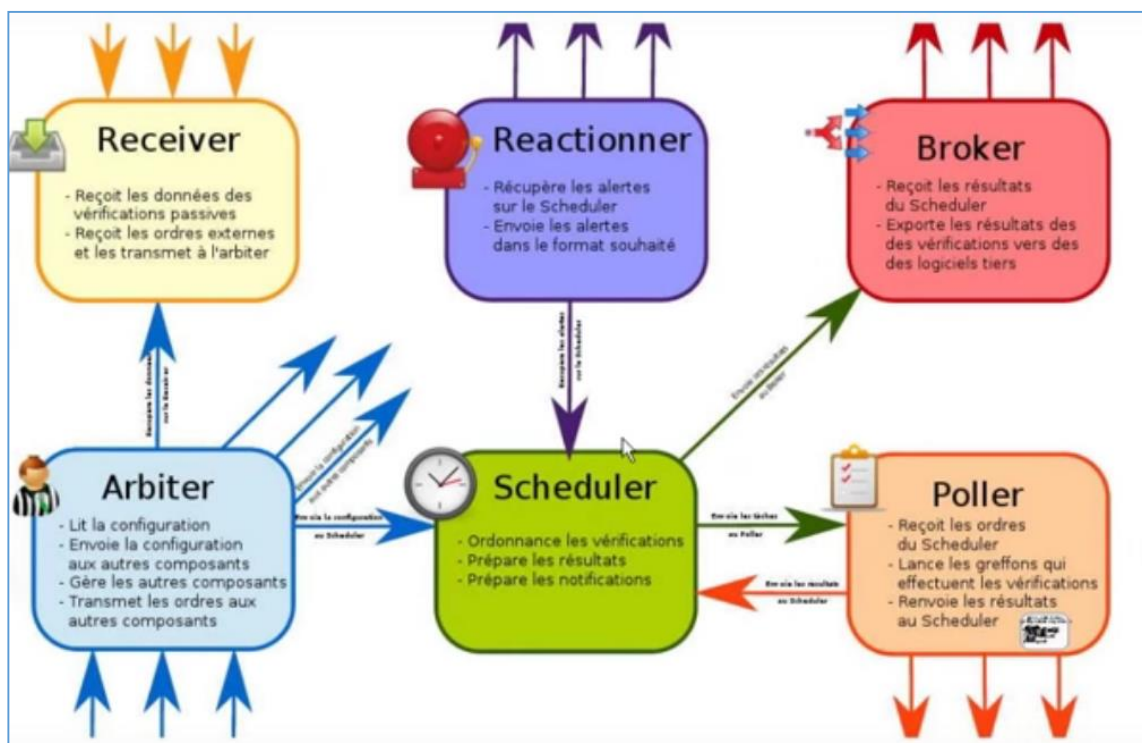


Figure III- 2: Les composants de Shinken

#### a. Arbitrer

L'arbitre (arbitre), c'est l'élément qui pilote tous les autres daemons, il se charge de la gestion et la transmission des ordres. Il lit et reçoit, la configuration, découpe cette dernière, en fichiers de configuration, pour dispatcher le travail, aux autres éléments. L'arbitre, surveille continuellement, les composants de l'architecture shinken (broker, reactionner, poller et scheduler).

## b. Scheduler

Le scheduler, a pour fonctionnement d'aménager, les vérifications et les checks, puis, examine et prépare, leurs résultats, ensuite, les données seront organisées dans une file d'attente. Une communication directe se déclenche avec, le poller, le broker et le reactionner. Le poller, pour les actions de control et de vérification, le broker, pour les actions de sauvegarde des informations et le reactionner pour les alertes et les notifications.

## c. Poller

Le poller, reçoit les requêtes du scheduler, puis, doit lancer les plugins et les greffons, pour effectuer les vérifications nécessaires. Enfin, les plugins, vont aller interroger le système surveillé, si un résultat est renvoyé par le plugin, il sera redirigé vers le scheduler.

## d. Reactionner

Le reactionner, se charge des notifications, selon les données du scheduler. Il récupère, les notifications et envoie des alertes sous le format souhaité, des emails ou des SMS d'alerte via des modules spécifiques. Il existe un module nommé Postfix, pour permettre à un serveur d'envoyer des e-mails sur une boîte de réception, et le module AndroidSMS, qui permet d'envoyer des SMS de notification, sur un téléphone mobile.

## e. Broker

Le broker, se charge d'interroger le scheduler, pour savoir, les données qui seront stockées. Ensuite il exporte les résultats, vers des logiciels. La sauvegarde des données, est effectuée via des modules de stockage spécifiés. L'injection les données, dans une source de stockage, sera par exemple dans, des fichiers texte, des bases de données ou des pages web,... Shinken récupère les informations depuis le broker, à l'aide d'une interface web intégrée, par le module WebUI.

## f. Receiver

Dans l'architecture de shinken, le receveur, est un élément facultatif, son rôle, est la réception des données passives, acquises d'un élément surveillé. Les données, seront sauvegardées temporairement dans un tampon.

## 2. Architecture globale du Shinken

Shinken, a une architecture modulaire et évolutive, cette dernière, supporte des grandes charges et fournit la haute disponibilité du système de supervision. À part le maître démon arbitre, nous pouvons multiplier et déployer les autres démons (scheduler, reactionner, broker et poller). La Figure III-3, ([19] s.d.) présente l'architecture globale de la solution shinken.

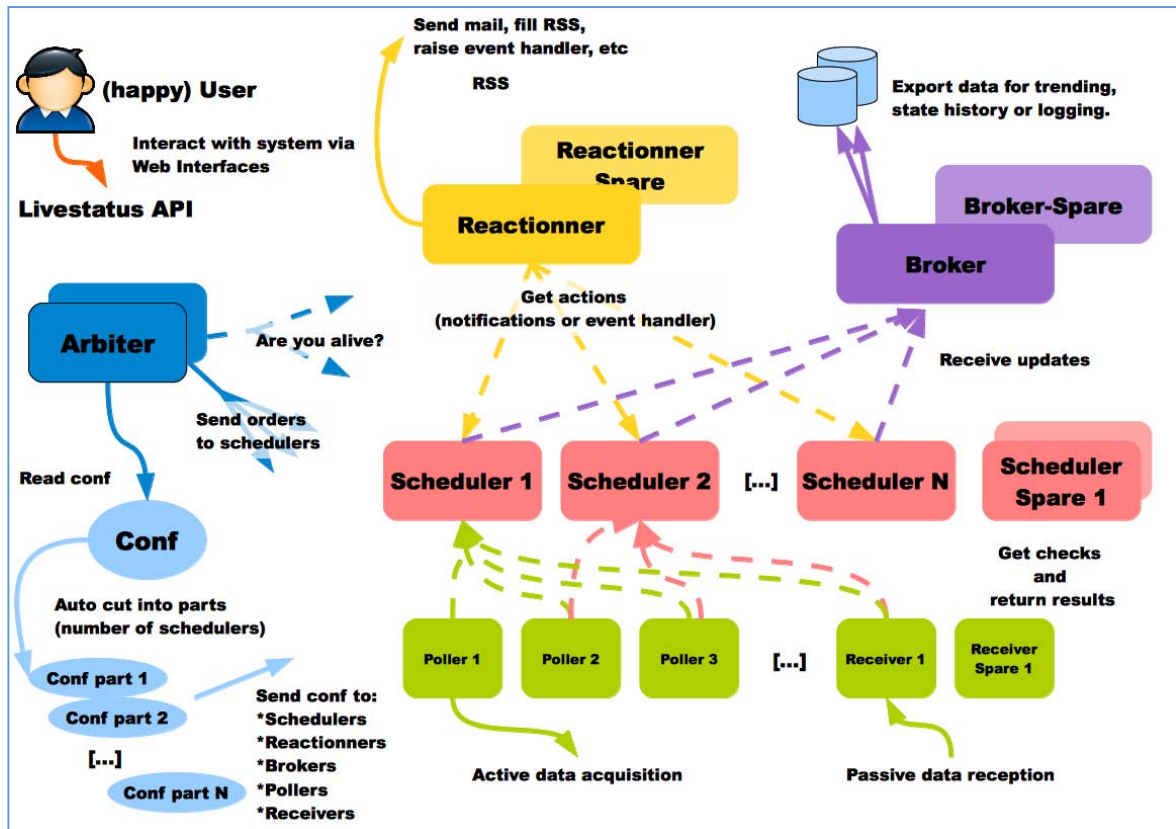


Figure III- 3 : Architecture globale shinken

Cette architecture, montre bien l'importance de la solution shinken, la haute disponibilité se manifeste, dans le cas de panne de l'un des daemons du système shinken. Si un élément devient indisponible, shinken, exécute un nouveau service, en utilisant un système de spare, qui est un système de backup, tout est prévu en cas de panne.

## III. Extensions à shinken

### 1. Les greffons

Les greffons, se chargent d'afficher du texte sur la sortie standard, via des exécutables lancés par le poller. Ils peuvent se présenter, sous forme de script ou bien des fichiers binaires compilés.

## 2. Les modules

Pour étendre, la fonctionnalité de la supervision, on utilise les modules, pour avoir des extensions aux daemons de shinken, tels que le broker et le receiver, qui fonctionnent uniquement avec les modules. Avec shinken, les modules, LiveStatus, WebUI et le module de log, sont des modules indispensables. L'interface graphique de shinken, est gérée par le module WebUI. [\\_\(\[20\] s.d.\)](#)

## 3. Plugins

Les plugins, sont des logiciels exécutables, effectuant une vérification et renvoyant, sur la sortie standard, un résultat, sous format texte. Les plugins, donnent des informations, en indiquant au serveur de supervision shinken, l'état des éléments surveillés, avec les codes de retours suivants :

- ✓ 0–OK : Hôte –UP, Tout va bien.
- ✓ 1–Warning : Risque potentiel problème
- ✓ 2 – Critical : Service UNKNOWN, Hôte DOWN ou UNREACHABLE Problème survenu
- ✓ 3 – Unknown : Service UNKNOWN, Hôte DOWN

## 4. Agents

Un agent, est un logiciel qui est installé sur un équipement supervisé, afin de fournir les données nécessaires au poller. L'approche actuelle est une supervision sans agent. Actuellement, la supervision, se fait via des programmes ou protocoles déjà installés comme SSH, SNMP,...Ces protocoles, permettent la récupération des informations directement, de la machine supervisée, puis la transmettre à un plugin de shinken. [\\_\(\[20\] s.d.\)](#).



La Figure III-4, \_([21] s.d.) illustre les flux de données échangés, entre les daemons, ainsi, les modules et les greffons utilisés par les acteurs shinken.

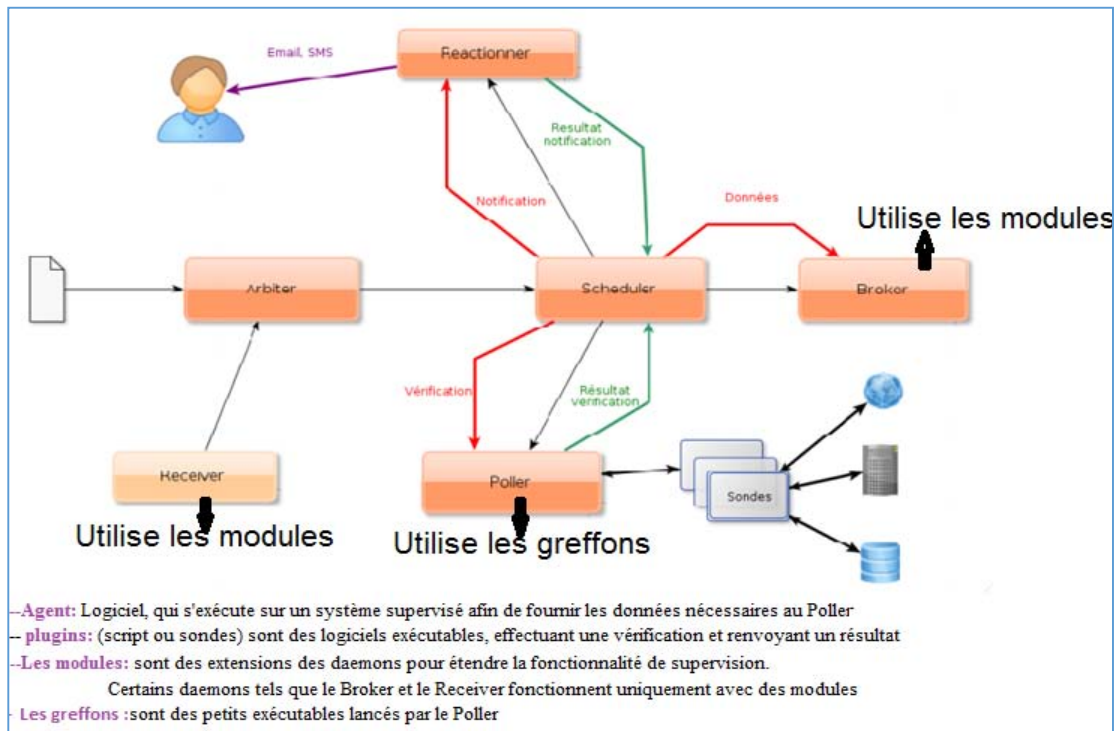


Figure III- 4: Modules/Greffons utilisés par les daemons

#### IV. Principe de fonctionnement du système shinken

Shinken utilise les daemons, qui ont chacun son rôle dans le système de supervision. Ces daemons se réunissent pour atteindre une saine supervision. Dans, ce qui suit, nous résumerons le scénario entre les acteurs de shinken :

- ✓ l'administrateur rentre sa configuration sur l'arbitrer, pour être découpée et dispatchée sur tous acteurs, et spécifiquement les schedulers.
- ✓ Les schedulers, vont planifier leurs contrôles, en fonction de la configuration qu'ils ont reçue.
- ✓ Les pollers, vont aller chercher les checks et les vérifications, qu'ils doivent être réalisés et exécutés via les plugins et les sondes. les résultats de vérifications, seront renvoyés aux schedulers.

A ce moment, le broker et le reactionner, peuvent commencer leurs travaux.

- ✓ Le reactionner va déclencher des notifications d'alerte (SMS, mail, ...) en fonction des résultats des contrôles, en interrogeant le scheduler.
- ✓ Ainsi, le broker va collecter les données du scheduler, il utilise des modules dédiés, comme, LiveStatus, MongoBD pour la gestion de la sauvegarde.



## Chapitre 4 : Spécification et conception des besoins

### Introduction

L'ingénierie de besoins, est l'une des activités importantes, parce qu'elle permet de déterminer, les besoins des utilisateurs d'une telle application et de définir les contraintes envisagées.

Dans ce chapitre, nous présenterons, la spécification et la conception des besoins, de notre solution shinken. D'abord, nous étudierons, les besoins fonctionnels, en présentant les modules de notre application, ainsi que les besoins non fonctionnels, en représentant les critères de l'environnement pour implémenter la solution shinken. Ensuite, nous ferons, une conception, en citant les acteurs de la solution et quelques diagrammes de cas d'utilisation.

### I. Spécification des besoins

#### 1. Les besoins fonctionnels

Les besoins fonctionnels, nous renseignent sur les fonctions principales traitées dans notre application, qui doit répondre à l'objectif de la supervision réseau. Notre objectif, est de mettre un serveur shinken, pour superviser le réseau du MAS. Notre solution, présente plusieurs fonctionnalités, qui se présentent sous forme de modules, qui seront traités dans ce qui suit.

##### a. Le module gestion du serveur shinken

Le module, gestion du serveur shinken, se caractérise par :

- ✓ L'authentification : L'utilisateur doit s'authentifier pour se connecter à l'interface de la solution en introduisant l'URL de son serveur et son login et mot de passe.
- ✓ La gestion des utilisateurs : La gestion d'utilisateurs nous permet d'ajouter, chercher, modifier ou supprimer un utilisateur.
- ✓ La gestion d'installation : Installer des modules et des plugins shinken (mise à jour, ajout de nouvelle fonctionnalité,...)

##### b. Le module de supervision des équipements

Le module de supervision des équipements, est formé des modules suivants :

- ✓ Gestion des équipements réseaux : Ajout, suppression, modification d'un équipement (routeur, Switch,...)

✓ Contrôler un équipement : contrôler une hôte consiste à consulter cette dernière pour collecter ou voir ses caractéristiques (taux d'utilisation de la mémoire, charge de CPU, espace disque,...)

✓ On peut aussi consulter un équipement, pour voir son état de fonctionnement UP ou DOWN

Pour contrôler les équipements informatiques, il faut consulter ces derniers, afin de collecter des informations, qui sont nécessaires au administrateur (ou superviseur), pour la bonne gestion des équipements et garantir une performance des interventions.

#### c. Le module gestion des pannes

Le module gestion des pannes, nous permet la détection, l'isolement de ces derniers, dans le but d'intervention et la maintenance des pannes, qui consiste à prendre rapidement les mesures correctives.

#### d. Le module générations des rapports

La solution shinken, permet la génération des rapports de chaque équipement réseau, qui décrit tous ses informations et ses caractéristiques, ainsi que les rapports d'évènements des pannes de chaque ressource.

#### e. Le module gestion des alertes

Ce module, permet la gestion des alertes et des notifications.

- ✓ Les alertes sont paramétrables via des plugins dédiées
- ✓ Les administrateurs, ont la charge d'acquittement des alertes et des notifications des pannes

## 2. Les besoins non fonctionnels

Pour répondre, aux contraintes liées à l'environnement et l'implémentation de la solution shinken, notre application doit répondre aux critères suivants :

✓ La sécurité : Shinken doit respecter la confidentialité des données et garantir l'intégrité de ces dernières, à chaque mise à jour et à chaque modification. L'accès aux données, nécessite une authentification par login et mot de passe.

✓ L'ergonomie de l'interface : Shinken, est disposé d'une interface facile et intégrée WebUI

- ✓ L'installation : installation facile
- ✓ La configuration : Shinken, exige une connaissance de Linux.
- ✓ L'extensibilité : Le système est extensible, il permet l'intégration des nouvelles fonctionnalités.
- ✓ La rapidité de traitement : Notre application doit être rapide dans l'exécution des traitements
- ✓ La performance : Notre application doit être performante à travers ses fonctionnalités, elle doit répondre à tous les besoins des administrateurs

## II. Le Langage de modélisation

Pour modéliser les fonctionnalités que doit offrir notre système et représenter les interactions entre ses différents composants, nous avons choisi le langage de modélisation unifié UML. UML, c'est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié », est un langage visuel constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. [\\_\(\[23\] s.d.\)](#)

Une vue est constituée d'un ou de plusieurs diagrammes pour représenter l'application (son fonctionnement, sa mise en place, ...). On distingue deux types de vue :

### 1. Les vues statiques

Ces vues, représentent le système physiquement, sous le format des diagrammes comme :

- ✓ Les diagrammes des classes
- ✓ Les diagrammes d'objet
- ✓ Les diagrammes de composants
- ✓ Les diagrammes de déploiements
- ✓ Les diagrammes de package,...

### 2. Les vues dynamiques

Ces vues, montrent le fonctionnement du système, sous le format des diagrammes comme :

- ✓ Les diagrammes d'activité
- ✓ Les diagrammes de cas d'utilisation
- ✓ Les diagrammes de séquence

- ✓ Les diagrammes de communication

### III. Conception et modélisation des besoins

Après avoir spécifié les différents besoins fonctionnels et non fonctionnels de notre projet, nous aborderons la partie analyse et modélisation des besoins. Cette partie consiste à faire une étude conceptuelle. Il s'agit, d'une part, de définir les acteurs de l'application, leurs rôles, ainsi que les différents diagrammes d'UML. Dans notre cas, nous nous serons limités, aux acteurs et quelques diagrammes de cas d'utilisation.

#### 1. Les acteurs

Un acteur, est toute entité qui interagit avec notre système, pour que l'acteur réponde à ses besoins. Ces acteurs sont l'administrateur et les superviseurs.

✓ L'administrateur : c'est le principal acteur auquel le système s'adresse, il est le responsable au bon fonctionnement de l'application de supervision, ses rôles sont :

- ✓ Gérer la plateforme et les comptes utilisateurs
- ✓ Etudier les hôtes, les services et les routeurs et commutateurs
- ✓ Communiquer avec la base de données du serveur
- ✓ Le superviseur : C'est l'acteur qui a comme rôle :
  - La consultation de l'état des hôtes
  - La consultation de l'état des équipements réseaux (Routeurs, Switch, etc.)
  - Et la consultation de l'état des services

#### 2. Diagrammes de cas d'utilisations

L'identification des cas d'utilisation, nous permet d'identifier les acteurs mis en jeu et visualiser les scénarios de notre application, pour bien superviser un équipement. Le diagramme de cas d'utilisation, représente les fonctionnalités en entités cohérentes, il décrit une vision globale de comportement fonctionnel d'une application.

## Diagramme de cas d'utilisation globale

La Figure IV-1, représente le diagramme de cas d'utilisation globale de la solution de supervision réseau.

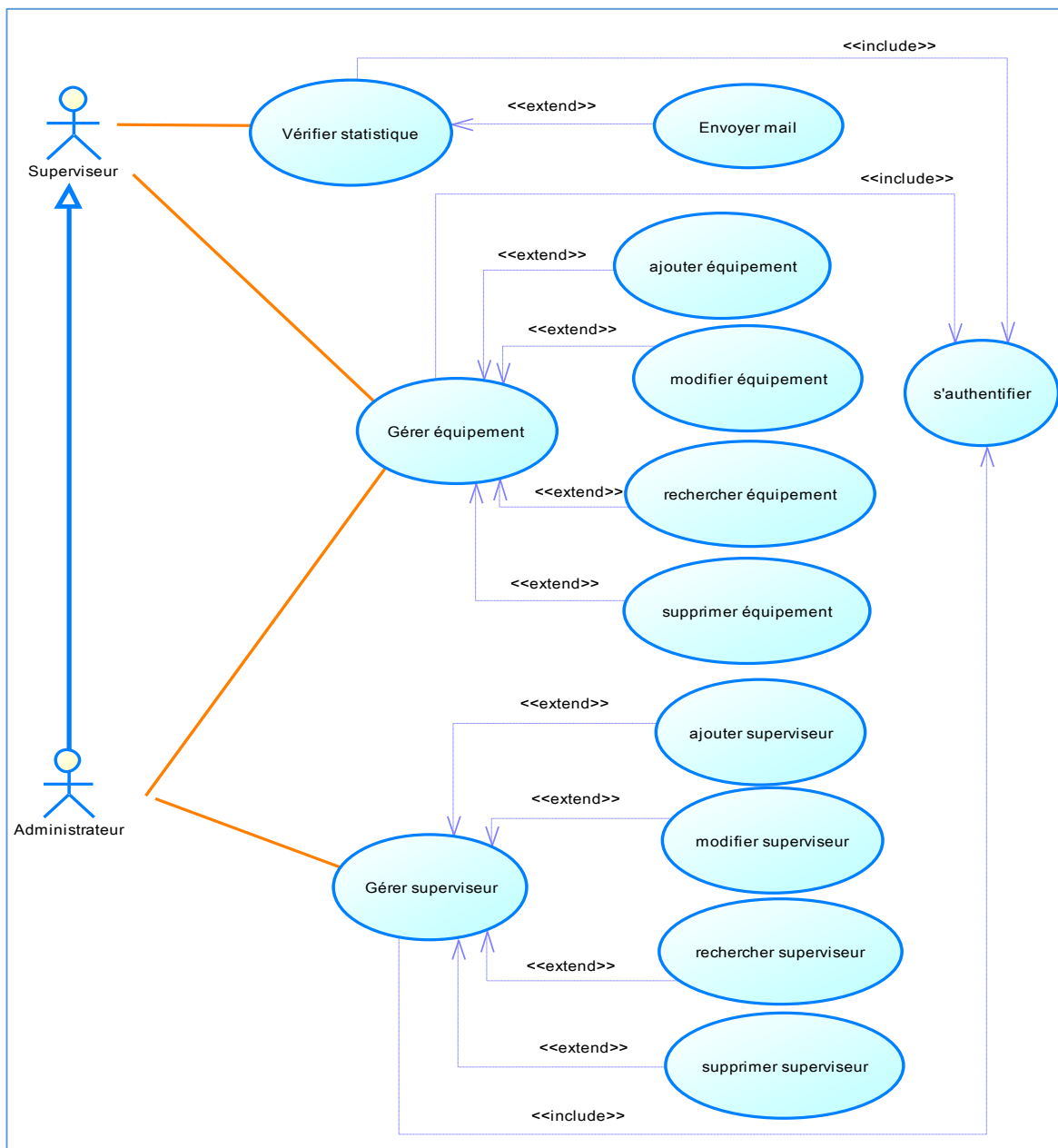


Figure IV- 1 : Diagramme de cas d'utilisation globale

### Diagramme de cas d'utilisation « Administrateur »

La Figure IV-2, représente le diagramme de cas d'utilisation, de l'acteur administrateur.

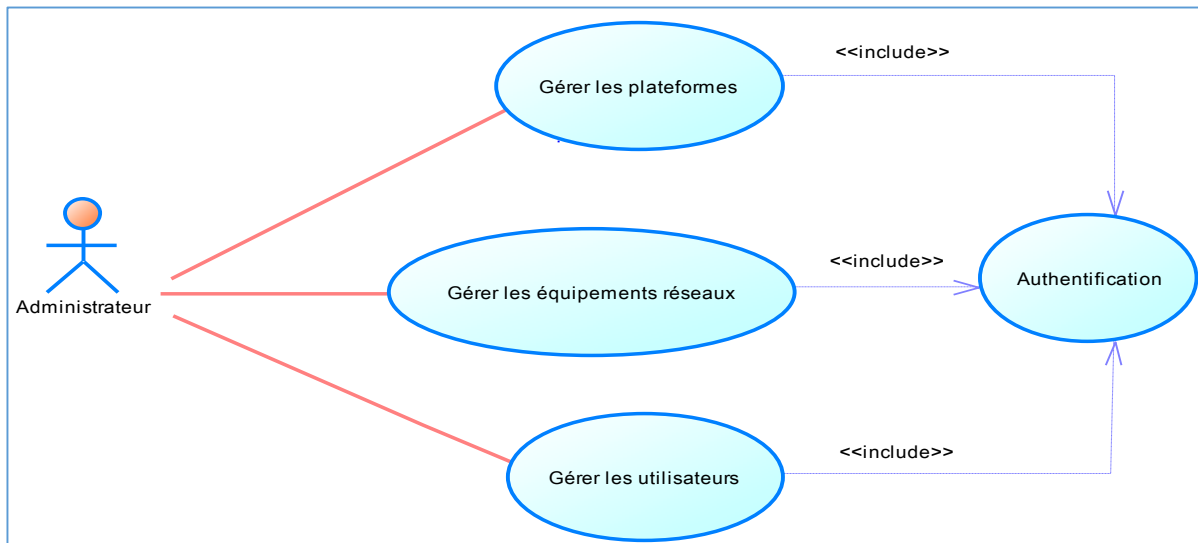


Figure IV- 2: Diagramme de cas d'utilisation « Administrateur »

### Diagramme de cas d'utilisation « Superviseur »

La Figure IV-3, représente le diagramme de cas d'utilisation, de l'acteur superviseur.

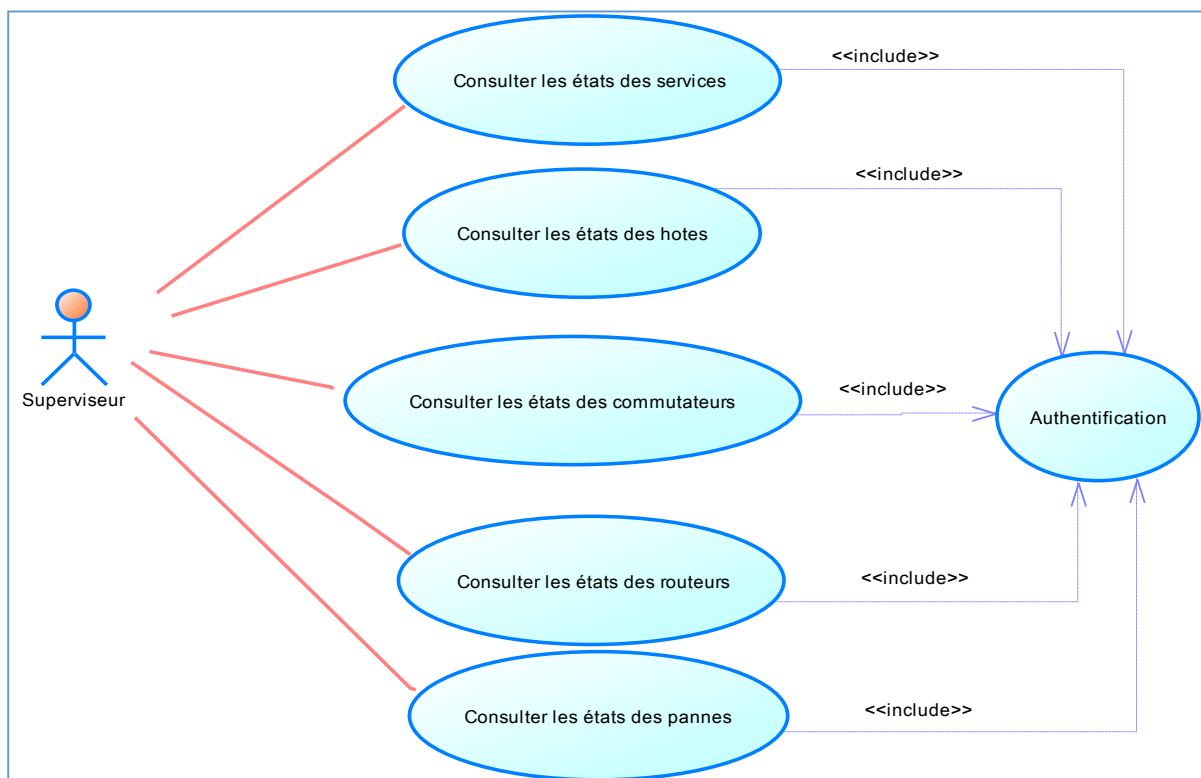


Figure IV- 3: Diagramme de cas d'utilisation « superviseur »



## Diagramme de cas d'utilisation « gestion des plateformes »

La Figure IV- 4, représente le diagramme de cas d'utilisation, du module gestion des plateformes.

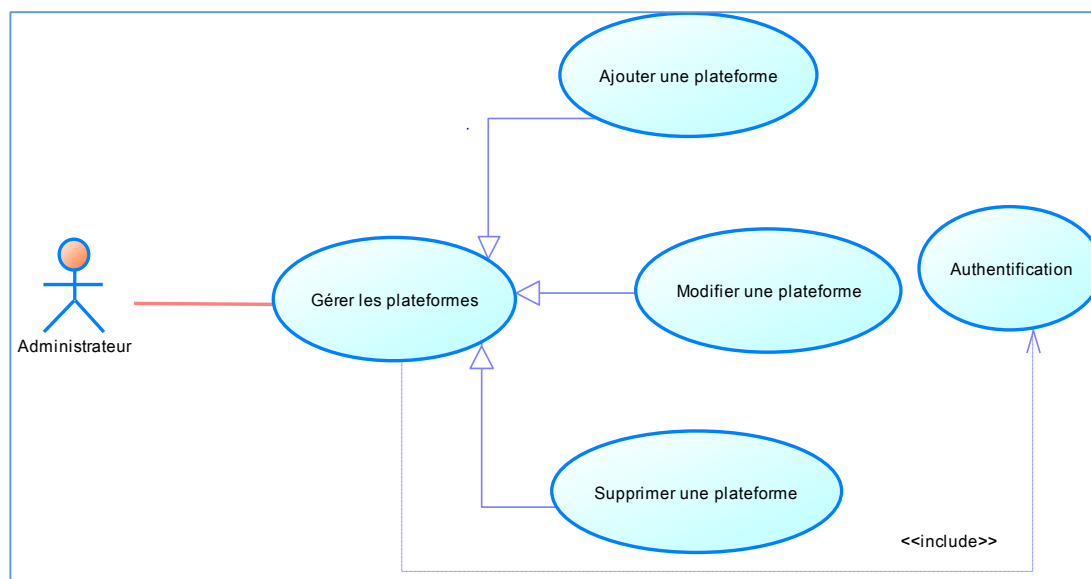


Figure IV- 4: Diagramme de cas d'utilisation « gestion des plateformes »

Nous présentons dans le Tableau IV-1, la description du diagramme de cas d'utilisation gestion des plateformes.

Tableau IV- 1 : Description de cas d'utilisation : gestion de plateforme

<b>Cas d'utilisation :</b> «gestion des plateformes».
<b>Résumé :</b> Gérer les plateformes
<b>Acteur :</b> Administrateur
<b>Pré condition :</b> Authentification
<b>Post condition :</b> Ajouter, modifier, consulter ou supprimer des plateformes.
<b>Scénario nominal :</b> <ol style="list-style-type: none"><li>1. Le système, affiche l'interface de la liste de plateformes.</li><li>2. L'utilisateur, fait le choix d'ajouter, modifier, consulter ou supprimer une plateforme</li><li>3. Le système vérifie le choix fournis</li><li>4. Le système donne l'interface convenable au choix de l'utilisateur</li><li>5. L'utilisateur remplit les champs donnés</li><li>6. L'utilisateur valide ou annule leurs saisies, l'interface de la liste des plateformes s'affiche</li></ol>

## Diagramme de cas d'utilisation « gestion des équipements réseaux »

La Figure IV- 5, représente le diagramme de cas d'utilisation, du module gestion des équipements.

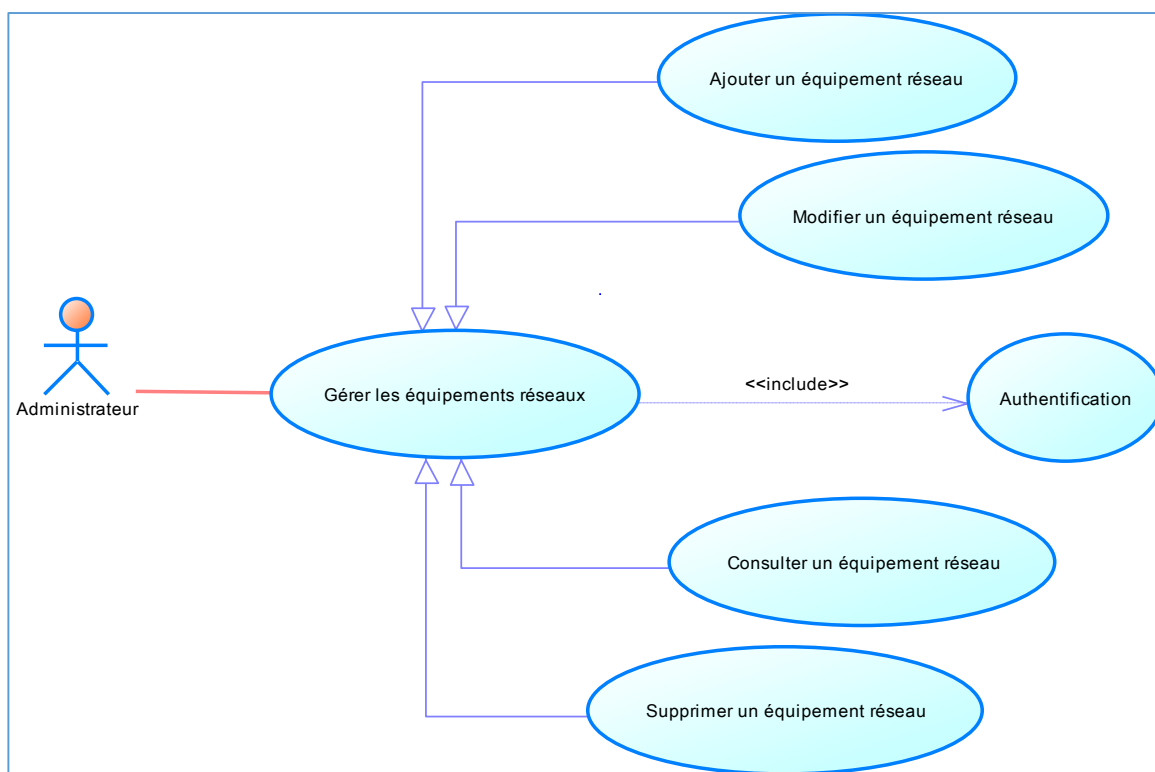


Figure IV- 5: Diagramme de cas d'utilisation « gestion des équipements réseaux »

Le Tableau IV-2, décrit explicitement le diagramme de cas, de la gestion des équipements réseaux

Tableau IV- 2: Description de cas d'utilisation « gestion des équipements réseaux »

<b>Cas d'utilisation :</b> «gestion des équipements réseaux»
<b>Acteur :</b> Administrateur.
<b>Résumé :</b> Gérer les équipements réseaux.
<b>Pré condition :</b> Authentification.
<b>Post condition :</b> Ajouter, modifier, consulter ou supprimer d'un équipement réseau
<b>Scénario principal :</b> <ol style="list-style-type: none"> <li>1. Le système affiche l'interface de la liste des équipements réseaux</li> <li>2. L'utilisateur fait son choix à propos des équipements réseaux</li> <li>3. Le système vérifie le choix fournis</li> <li>4. Le système donne l'interface d'ajouter, modifier, consulter ou supprimer</li> <li>5. L'utilisateur fait son choix et fini par validation.</li> </ol>

## Diagramme de cas d'utilisation « gestion des utilisateurs »

La Figure IV- 6, représente le diagramme de cas d'utilisation, du module, gestion des utilisateurs.

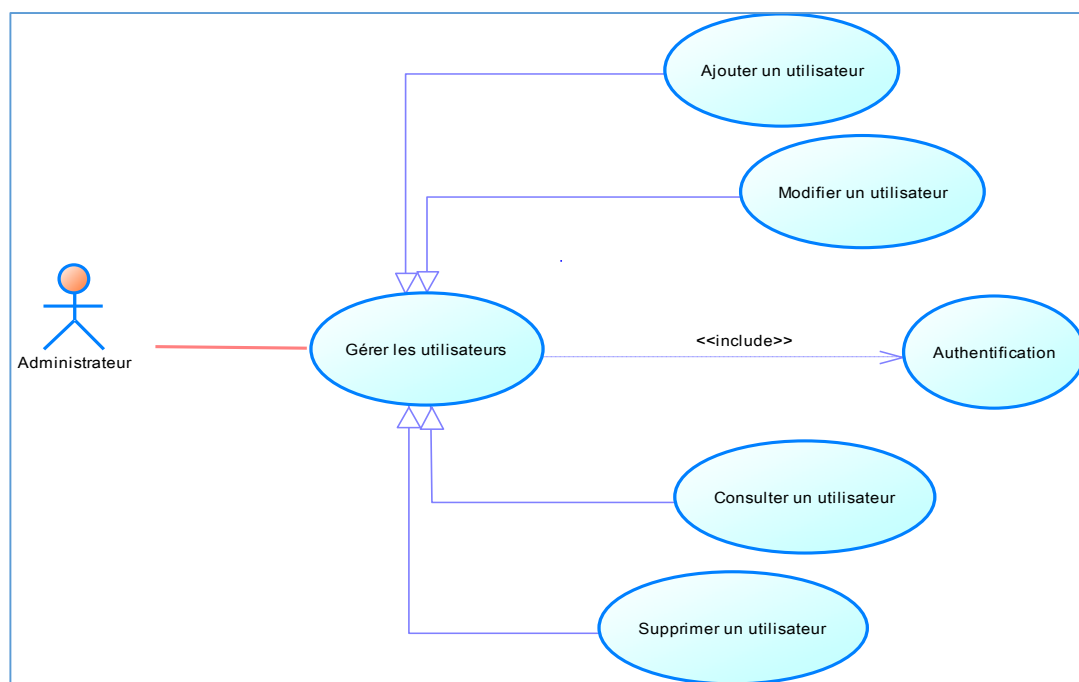


Figure IV- 6: diagramme de cas d'utilisation « gestion des utilisateurs »

La description de ce diagramme, est bien détaillée dans le Tableau IV-3.

Tableau IV- 3 : Cas d'utilisation «gestion des utilisateurs».

<b>Cas d'utilisation</b> : «gestion des utilisateurs».
<b>Résumé</b> : Gérer les utilisateurs
<b>Acteur</b> : Administrateur
<b>Pré condition</b> : Authentification
<b>Post condition</b> : Ajouter, modifier, consulter ou supprimer des utilisateurs.
<b>Scénario principal</b> : <ol style="list-style-type: none"><li>1. Le système affiche l'interface de la liste des utilisateurs</li><li>2. L'utilisateur fait son choix pour les utilisateurs</li><li>3. Le système ouvre la page des paramètres</li><li>4. L'utilisateur saisit leurs données</li><li>5. L'utilisateur valide leurs choix pour être rediriger vers la page principale</li></ol>

## Conclusion

Dans ce chapitre, nous avons présenté, la spécification et la conception des besoins, de la solution de supervision réseau shinken. D'abord, nous avons étudié, les besoins fonctionnels, en présentant les modules et ses fonctions principales traitées dans l'application. Ensuite, nous avons fait, une conception des besoins, en introduisant le langage de modélisation UML. Enfin, nous avons cité les acteurs intervenants, dans la gestion de la solution de supervision réseau, avec l'étude de quelques diagrammes de cas d'utilisation.

## Chapitre 5 : Mise en place de la solution shinken

### Introduction

Tout d'abord, nous présenterons l'architecture réseau sur laquelle nous développerons notre solution, en définissant l'environnement matériel les logiciels utilisés. Ensuite, nous nous occuperons de la mise en place de la solution, en préparant le serveur, l'installation et la configuration de shinken. Enfin nous présenterons notre réalisation en détaillant quelques captures d'écrans représentant, le fonctionnement du système de supervision réseau shinken.

### I. Architecture du travail

L'architecture réseau sur laquelle nous allons développer notre application au sein du ministère des affaires sociales, est représentée dans la Figure V-1, qui comporte les directions liées à cette dernière dans la grande Tunis (Beb-Bnet1, Beb-Bnet2, Parc-Auto, Beb-El-khadra, Dar-Assabeh et Rue de Bayrouth).

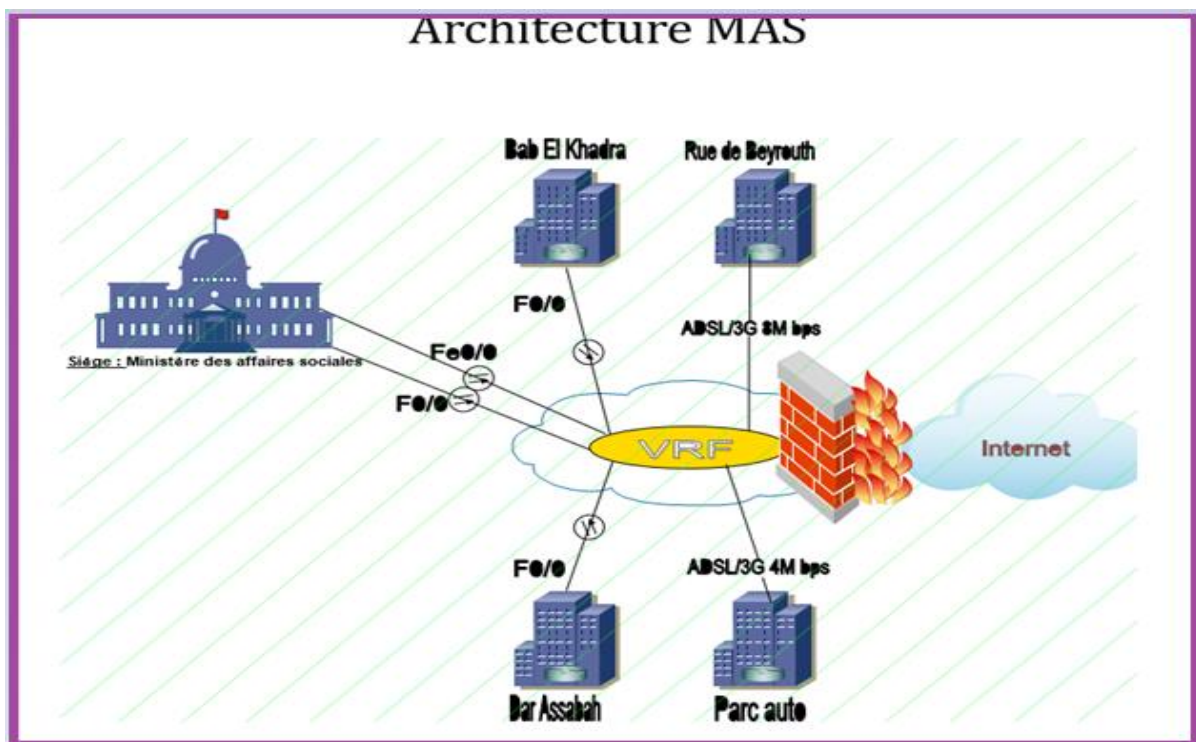


Figure V- 1 : Architecture de travail

#### 1. Environnement matériel

L'environnement matériel de notre application de supervision est composé de :

Six routeurs : un routeur, est un équipement de couche 3, par rapport au modèle OSI, il permet le routage des paquets.

- ✓ Type : Routeurs Cisco

Un serveur : Serveur de supervision shinken

- ✓ UC : Dell Processeur Intel Core i3
- ✓ Mémoire : 4Go de RAM / DD 500GO
- ✓ Debian 8

Un ordinateur :

- ✓ Windows 7 64bits

## 2. Environnement logiciel :

Putty : est un émulateur de terminal dédié pour le protocole SSH qui permet d'établir une connexion directe avec le serveur de supervision.

WinsCP : est un client graphique open source pour Windows. Il peut réaliser toutes les opérations de base sur les dossiers et les fichiers entre un ordinateur local et un ordinateur distant.

## II. Mise en place de la solution

Cette phase consiste à mettre en place le serveur shinken au sein du ministère des affaires sociales.

### 1. Préparation du serveur shinken

#### a. Installation du Debian 8 sur le serveur Shinken

Premièrement, nous avons installé Debian 8 et configuré l'adresse réseau via la commande :

```
vi /etc/network/interfaces  
inet static  
address xx.xx.xx.ip  
netmask xxx.xxx.xxx.xxx  
gateway  
dns-name-servers  
/etc/init.d/networking restart
```

Puis, le serveur est branché dans la salle de serveur du ministère des affaires sociales et nous l'administrons à distance.

b. Installation de WinsCP sur la machine Administrateur

L'installation et la configuration sont détaillées dans « l'[Annexe B](#) »

c. Configuration de Putty sur la poste Administrateur

La configuration est détaillée dans « l'[Annexe B](#) »

2. Installation de Shinken :

a. Téléchargement et installation des mises à jour

Le premier pas à faire est l'installation des mises à jour pour obtenir les dernières versions qui nous permettons d'installer convenablement le serveur shinken, donc il faut les téléchargées et les installées via la commande depuis le root :

```
apt-get update && sudo apt-get upgrade
```

b. Installation des dépendances nécessaires :

Shinken est une suite de paquets à installer dans une distribution sous Linux, que ce soit Debian (pour notre cas), Ubuntu, ... Les dépendances des hôtes et des services sont une fonctionnalité avancée qui nous permet de contrôler le comportement des hôtes et des services selon l'état d'un ou plusieurs autres hôtes ou services via la commande :

```
apt-get install python-pip python-pycurl python-cherrypy3 python-setuptools python-crypto -y
```

✓ Les daemons de Shinken seront plus performants si nous installons la librairie python-cherrypy3

✓ python-setuptools : est une librairie et un outil de distribution de code Python, et elle permet notamment d'installer des librairies externes automatiquement en les téléchargeant depuis la librairie python-setuptools.

✓ En utilisant l'outil « pip » qui permet de récupérer des package python du site Pypi. Pypi, est le dépôt du langage de programmation Python. Son objectif est de donner aux développeurs Python d'un catalogue complet contenant tous les paquets Python libres.

L'installation, se fait via la commande :

```
#pip install pymongo>=3.0.3 requests arrow bottle==0.12.8
```

c. Installation des packages python

Télécharger shinken :# `wget https://github.com/naparuba/shinken/tarball/master`

`#tar -xvzf master`

Se loguer sur :

`cd naparuba-shinken-37f8670`

`python setup.py install`

d. Création d'un utilisateur shinken :

Il est aussi nécessaire de créer un utilisateur "shinken" pour permettre l'installation et l'utilisation du serveur shinken via la commande :

`#adduser shinken`

Une fois la création d'utilisateur shinken est faite et l'installation des dépendances nécessaires sont réalisées, on aura l'arborescence du répertoire `/etc/shinken` de la Figure V-2.

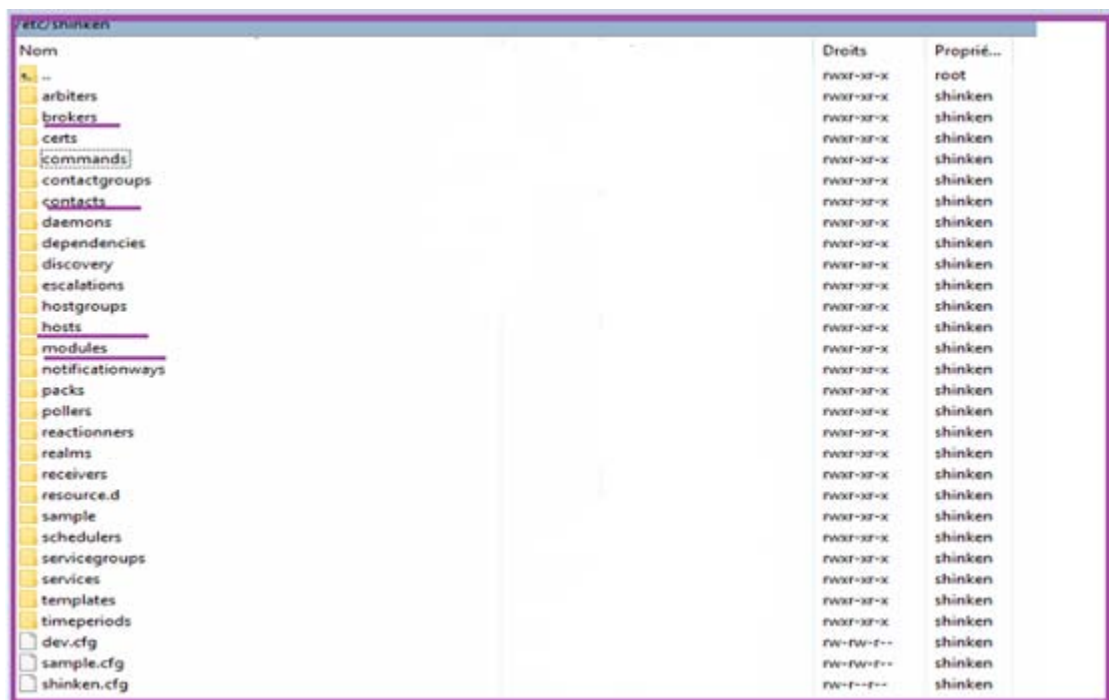


Figure V- 2 : Le répertoire `/etc/shinken`

Le dossier « `/etc/shinken` » est créé par défaut lors de l'installation de shinken. Là où se trouvent les fichiers de configuration du serveur en lui-même, ainsi que les commandes



disponibles et les fichiers concernant les composants de shinken (arbiter, scheduler, poller, reactionner et broker).

/usr/bin/shinken-\*: contient tous les daemons et les scripts de shinken

/var/lib/shinken : contient toutes les modules, les check et les plugins de shinken.

/usr/bin/shinken et /var/lib/shinken, contiennent tous les fichiers nécessaires au bon fonctionnement de shinken.

/var/log/shinken : les divers évènements des composants de shinken, les erreurs et les bugs possibles de la supervision, sont répertoriés dans des fichiers logs du répertoire /var/log/shinken.

#### e. Démarrage de shinken

Après la création de l'utilisateur shinken, tous les daemons sont lancés. Maintenant nous pouvons superviser notre réseau. En premier lieu, il faut se loguer sous shinken.

**#su shinken**

Shinken, a besoin d'être initialisé lors de la première utilisation afin de générer le fichier `~/shinken.ini` contenant les chemins vers les différents répertoires de configuration de l'outil.

Pour ça on lance la commande : **shinken - -init**

#### f. Installation des packs nécessaire pour le serveur shinken

Il faut installer les différents packs permettant de remonter l'information des hôtes distants vers le serveur shinken. Une hôte, est un élément réseau (serveurs, postes de travail, switch, routeur,...) dont les services (disque, mémoire, cpu, ...) vont être supervisés par le serveur Shinken.

Ajout du pack linux-snmp : Pack pour le check en snmp

**shinken install linux-snmp**

Ajout du pack routeur : Pack pour check de routeur

**shinken install router**

Ajout du pack Cisco : Pack pour le check des équipements Cisco

**shinken install cisco**

Ajout du pack switch : Pack pour le check de switch

## shinken install switch

```
shinken@shinken:~$ shinken install linux-snmp
Grabbing : linux-snmp
OK linux-snmp
shinken@shinken:~$ shinken install router
Grabbing : router
OK router
shinken@shinken:~$ shinken install cisco
Grabbing : cisco
OK cisco
shinken@shinken:~$ shinken install switch
Grabbing : switch
OK switch
shinken@shinken:~$ █
```

Installation de l'interface web Webui2 :

L'interface web webui2, est un module du daemon broker, qui s'occupe de la lecture, de l'interprétation et de l'affichage des résultats obtenus dans les fichiers logs. L'installation de webui2, se fait depuis l'utilisateur shinken via la commande :

## \$shinken install webui2

```
root@shinken:/home/saida# su shinken
shinken@shinken:/home/saida$ shinken install webui2
Grabbing : webui2
OK webui2
shinken@shinken:/home/saida$ █
```

## Remarque

À la fin de l'installation de chaque outil sous shinken, il faut redémarrer les services shinken `/etc/init.d/shinken restart`, le résultat est :

```
Restarting scheduler
. ok
Restarting poller
. ok
Restarting reactionner
. ok
Restarting broker
. ok
Restarting receiver
. ok
Restarting arbiter
Doing config check
. ok
. ok
```

g. Connexion à l'interface shinken :

Maintenant on peut accéder à l'interface web de notre serveur à partir de l'adresse <http://IP-SERVEUR:7767>. La Figure V-3, représente l'interface de connexion au serveur-shinken, dans un navigateur internet (Firefox par exemple)

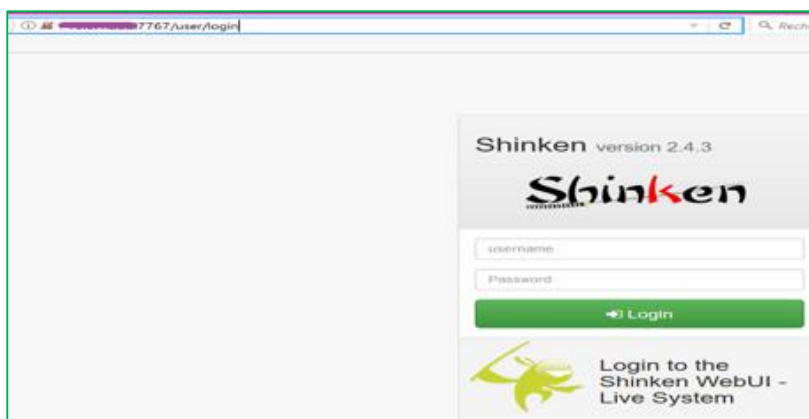


Figure V- 3 : Interface de connexion

Les utilisateurs sont déclarés dans /etc/shinken/contacts/, par défaut l'administrateur, sa configuration est dans le fichier admin.cfg représenté par la Figure V- 4, le login et le mot de passe sont admin/admin.

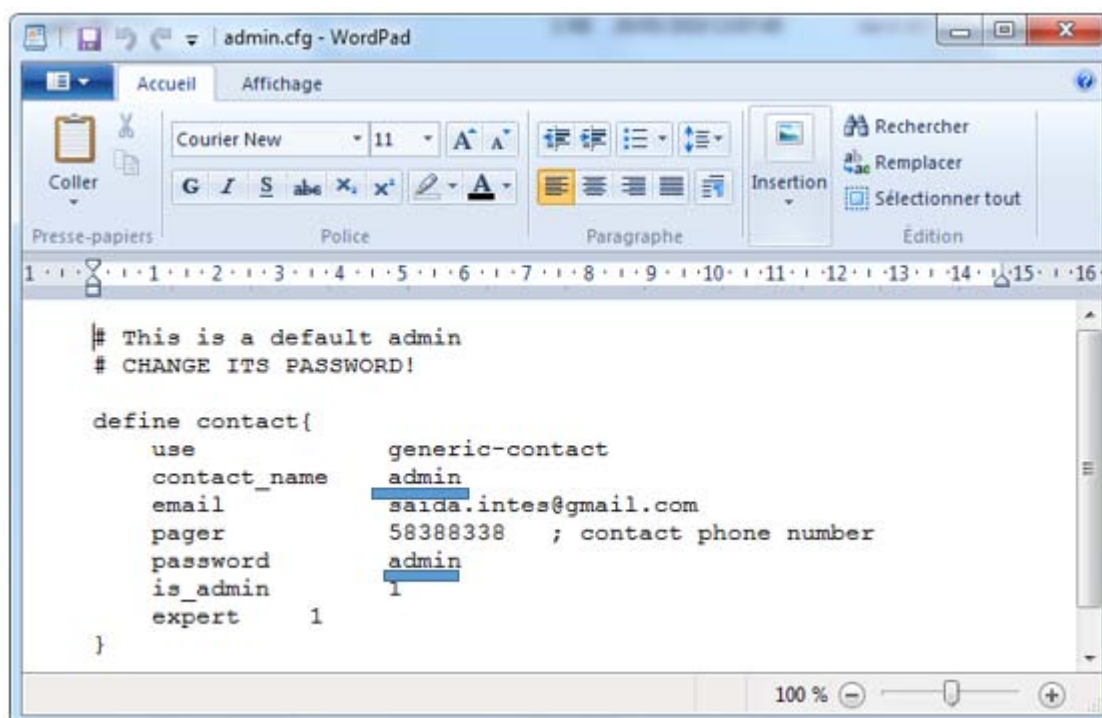


Figure V- 4 : Le fichier admin.cfg

### 3. La configuration du serveur-shinken

La configuration du serveur shinken, l'interface shinken WebUI, le login et le mot de passe, et le matériel de l'architecture de travail (switch, routeur), est plus détaillée dans « l'[Annexe C](#) »

#### 4. Observation de l'état du serveur et des routeurs :

La Figure V-4 et la Figure V-5, montrent l'état du serveur shinken et les routeurs qui sont UP, ainsi les états des services (Ok, Warning, Critical, Unknown).

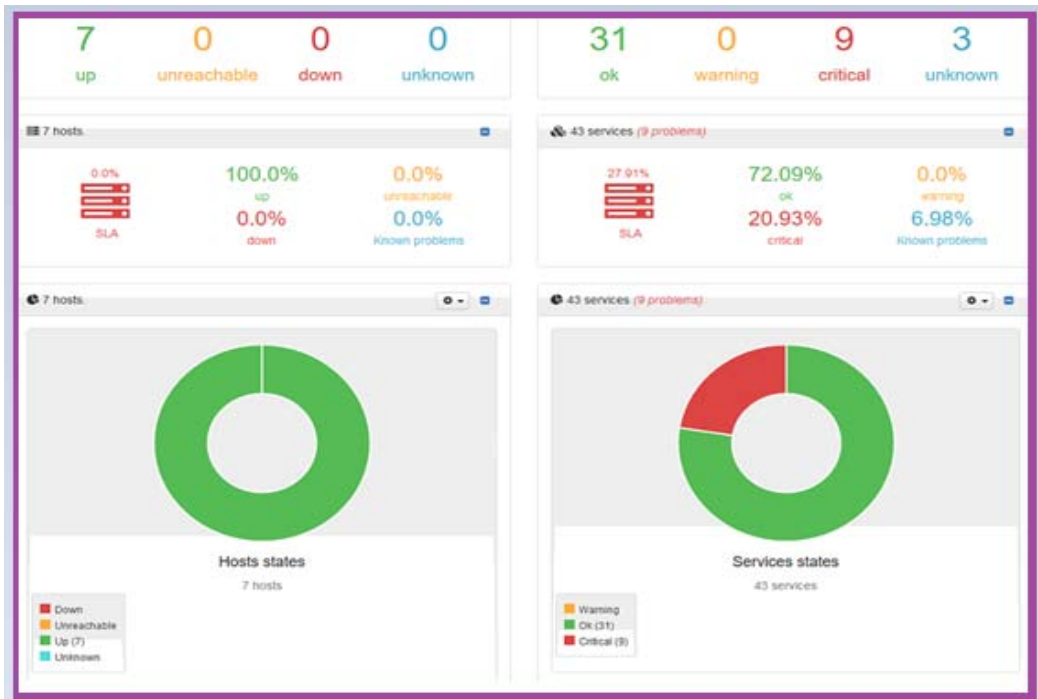


Figure V- 5 : Etat du serveur shinken et des routeurs

La Figure V-6, montre les pourcentages des états des services

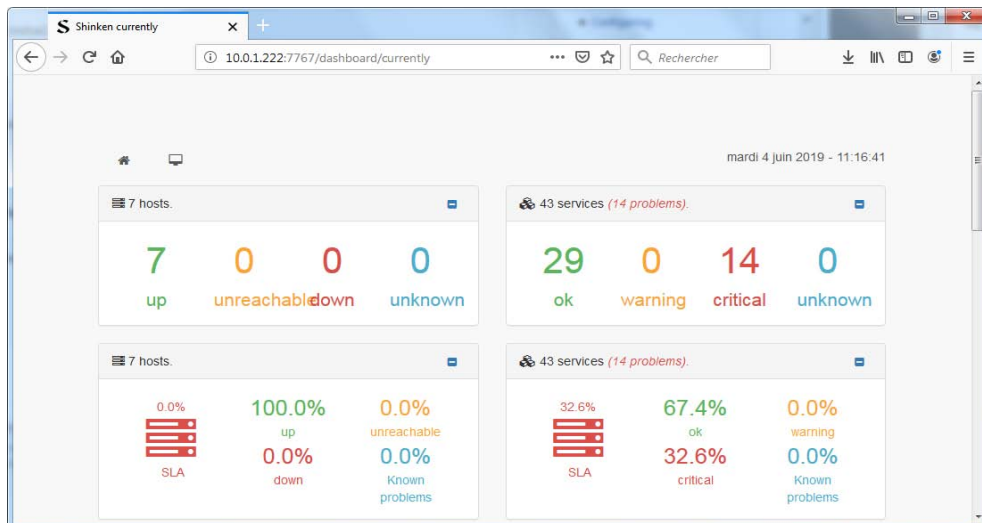


Figure V- 6 : Etat du serveur shinken et des routeurs

La Figure V-7, montre les services qui sont OK de chaque équipement réseau.

- CPU
- Hardware
- InterfaceUsage
- Memory,...

Device	Service	Status	Uptime	Description
R-Beb-Bnet-1	Cpu	OK	3h 27m	OK - cpu 1 usage (5 min avg.) is 2.00%
	Hardware	OK	3h 23m	OK - environmental hardware working fine
	InterfaceErrors	OK	3h 27m	OK - interface Embedded-Service-Engine0/0 errors i...
	InterfaceUsage	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 usage i...
	Memory	OK	3h 24m	OK - mempool Processor_1 usage is 14.33%, mempo...
R-Beb-Bnet-2	Cpu	OK	3h 22m	OK - cpu 1 usage (5 min avg.) is 1.00%
	Hardware	OK	3h 25m	OK - environmental hardware working fine
	InterfaceErrors	OK	3h 26m	OK - interface Embedded-Service-Engine0/0 errors i...
	InterfaceUsage	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 usage i...
	Memory	OK	3h 24m	OK - mempool Processor_1 usage is 14.34%, mempo...
R-Rbh-Khadhra	Cpu	OK	3h 22m	OK - cpu 1 usage (5 min avg.) is 1.00%

Figure V- 7 : Etat des services

Pour d'autres graphes du fonctionnement du serveur shinken, voir « [Annexe D](#) »

## 5. Notification

Pour que notre messagerie fonctionne, il faut installer le paquet html, qui permet l'envoi des messages générées par les scripts de shinken.

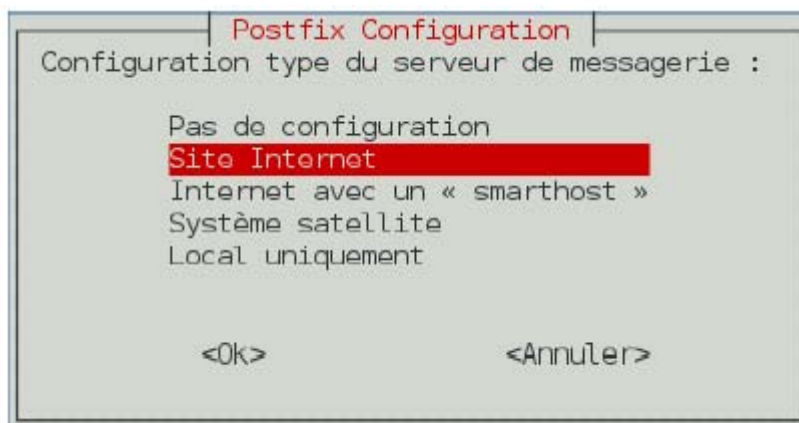
```
root@Test-Shinken:/# pip install html
Downloading/unpacking html
  Downloading html-1.16.tar.gz
  Running setup.py (path:/tmp/pip-build-ihGI1o/html/setup.py) egg_info for package html

Installing collected packages: html
  Running setup.py install for html

Successfully installed html
Cleaning up...
root@Test-Shinken:/#
```

Installation et configuration du service de messagerie Postfix,

`#apt-get install postfix`



Le nom de notre courrier, est Serveurshinken

```
Postfix is now set up with a default configuration.  If you need to make
changes, edit
/etc/postfix/main.cf (and others) as needed.  To view Postfix configuration
values, see postconf(1).

After modifying main.cf, be sure to run '/etc/init.d/postfix reload'.
```

Ensuite, on redémarre le service postfix : `#service postfix restart`

Nous pouvons maintenant tester l'envoi de notre messagerie, pour ce faire, nous allons envoyer un mail à shinken@serveurshinken, de la manière suivante :

`mail -s "objet du message" adresse-mail du destinataire et le contenu`

Exemple :

```
.shinken@Test-Shinken:~$
shinken@Test-Shinken:~$ mail -s 'Test' shinken@Test-Shinken
Cc: Null message body; hope that's ok
Vous avez du nouveau courrier dans /var/mail/shinken
.shinken@Test-Shinken:~$
```

Quitter le message, nous taperons ctrl+d et pour l'envoyer ctrl+d une autre fois.

Nous pouvons consulter notre boîte mail.

```
shinken@Test-Shinken:~$ mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/shinken": 3 messages 3 new
>N 1 root@Test-Shinken  Sun Jun  2 19:14   12/443  Test
  N 2 shinken@Test     Sun Jun  2 22:13   12/463  Test
  N 3 shinken@Test     Sun Jun  2 22:14   12/463  Test
&
Message 1:
From root@Test-Shinken  Sun Jun  2 19:14:44 2019
X-Original-To: shinken@Test-Shinken
To: shinken@Test-Shinken
Subject: Test
Date: Sun,  2 Jun 2019 19:14:44 +0100 (CET)
From: root@Test-Shinken (root)

&
Message 2:
From shinken@Test-Shinken  Sun Jun  2 22:13:04 2019
X-Original-To: shinken@Test-Shinken
To: shinken@Test-Shinken
Subject: Test
Date: Sun,  2 Jun 2019 22:13:04 +0100 (CET)
From: shinken@Test-Shinken

&
Message 3:
From shinken@Test-Shinken  Sun Jun  2 22:14:26 2019
X-Original-To: shinken@Test-Shinken
To: shinken@Test-Shinken
Subject: Test
Date: Sun,  2 Jun 2019 22:14:26 +0100 (CET)
From: shinken@Test-Shinken

&
At EOF
```

Configuration des notifications shinken.

Le fichier, /etc/shinken/notificationways/email.cfg, est de la forme suivante

```
# This is how emails are sent, 24x7 way.
define notificationway{
    notificationway_name      email
    service_notification_period 24x7
    host_notification_period   24x/
    service_notification_options c,w,r
    host_notification_options  d,u,r,f,s
    service_notification_commands notify-service-by-email; send
service notifications via email
    host_notification_commands notify-host-by-email; send
host notifications via email
}
```

Les courriers, sont envoyés 24 heures sur 24 pendant les sept jours de la semaine. Les notifications, nous renseigne sur l'état des hôtes et des services.

### Le temps des alertes

Le fichier `/etc/shinken/timeperiods/24x7.cfg`, nous renseigne sur le temps des alertes, nous recevons ces derniers, tout le temps, tous les jours, du lundi au dimanche et pendant 24 heures (00:00-24:00).

```
define timeperiod{
    timeperiod_name    24x7
    alias              Always
    -sunday            00:00-24:00
    -monday            00:00-24:00
    -tuesday           00:00-24:00
    -wednesday         00:00-24:00
    -thursday          00:00-24:00
    -friday            00:00-24:00
    -saturday          00:00-24:00
}
```

### Ajouter l'utilisateur shinken au contactgroups

Notre utilisateur shinken, doit être ajouté dans le fichier `/etc/shinken/contactgroups` qui contient les membres du groupe admins,

```
root@Test-Shinken:/home/shinken# cat /etc/shinken/contactgroups/admins.cfg
define contactgroup{
    contactgroup_name    admins
    alias                admins
    members               admin, shinken
}
root@Test-Shinken:/home/shinken#
```

### La fréquence d'envoi des notifications

Le fichier, `/etc/shinken/packs/linux-snmp/services/disks.cfg`, nous donne la fréquence à laquelle il envoie des notifications. Par défaut, il envoie des notifications à chaque heure.



/etc/shinken/packs/linux-snmp/services/				
Nom	Taille	Date de modification	Droits	Proprié...
..		27/06/2014 14:27:43	rxr-xr-x	root
time.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root
network_usage.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root
memory.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root
logFiles.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root
load.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root
disks.cfg	1 KB	03/06/2019 00:44:36	rw-r--r--	root
cpu.cfg	1 KB	27/06/2014 14:27:43	rw-r--r--	root

```

define service {
    service_description    Disks
    use                    1hour_long,linux-service
    register                0
    host_name              linux-snmp
    check_command          check_linux_disks

    _DETAILLEDESC         Overall disks usage
    _IMPACT                Depends on disks, cause system instability
    _FIXACTIONS            Clean the appropriate disks
}

```

Nous pouvons modifier le temps, nous choisissons le modèle qui nous intéresse, par exemple la fréquence 5 minutes.

```

define service {
    service_description    Disks
    use                    5min_short,linux-service
    register                0
    host_name              linux-snmp
    check_command          check_linux_disks

    _DETAILLEDESC         Overall disks usage
    _IMPACT                Depends on disks, cause system instability
    _FIXACTIONS            Clean the appropriate disks
}

```

```
shinken@Test-Shinken:~$ cat /etc/shinken/packs/linux-snmp/services/disks.cfg
define service {
    service_description    Disks
    use                    5min_short,linux-service
    register                0
    host_name              linux-snmp
    check_command          check_linux_disks

    _DetaIleDESC          Overall disks usage
    _IMPACT                Depends on disks, cause system instability
    _FIXATIONS            Clean the appropriate disks
}
shinken@Test-Shinken:~$
```

## Conclusion

Dans ce chapitre, tout d'abord, nous avons étudié l'architecture réseau sur laquelle nous avons implémenté notre solution, avec une description de l'environnement matériel et logiciels utilisés. Ensuite, nous avons occupé de la mise en place de la solution, avec la préparation du serveur, l'installation et la configuration de shinken. Enfin, nous avons présenté notre solution, via quelques captures d'écrans qui représentent, le système de supervision réseau shinken et qui résumant l'état de fonctionnement des équipements de notre architecture.

## Conclusion Générale

Les années dernières, nous a prouvé que la recherche de la fiabilité et de la réduction des coûts fait apparaitre l'avenir du monde informatique dans le domaine de la supervision dont on parle des applications open source.

Un outil de supervision d'un système informatique, offre une visibilité claire et simple à interpréter. En effet ces outils permettent de vérifier ce qu'il se passe à tout moment au niveau du réseau, et nous autorisent à réagir rapidement en cas de problème.

Durant ce stage, nous avons appris à mettre en place une solution de surveillance par l'installation d'un serveur de supervision shinken dans le ministère des affaires sociales. Cette excellente expérience nous a amené à faire beaucoup de recherche et avoir une visibilité concrète sur un domaine bien spécifique, qui est la supervision d'un réseau informatique. En plus, ce travail nous a permis d'acquérir une expérience en Linux, Ubuntu et Debian, à travers laquelle nous avons eu l'occasion d'appliquer nos connaissances et de confronter et appliquer la notion théorique à la pratique sur le terrain dans le domaine réseau informatique.

Comme perspectives, nous proposons l'amélioration de ce travail par :

- ✓ La configuration des notifications par SMS.
- ✓ La supervision des applications comme les bases de données.
- ✓ La supervision des équipements du parc informatique (ordinateurs, imprimantes,...)
- ✓ Intégration de shinken avec GLPI (Gestion libre de parc informatique)

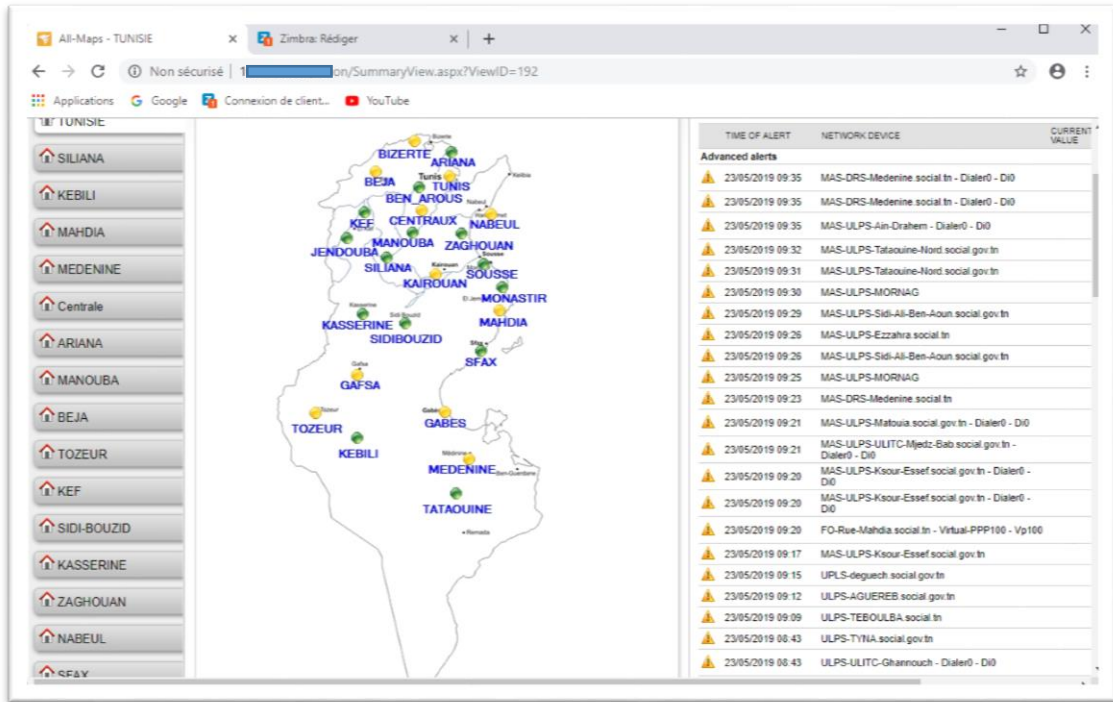
## Webographie

- [1]. s.d. [http://www.legislation.tn/detailtexte/D%C3%A9cret-num-1996-269-du----jort-1996-017\\_\\_1996017002693](http://www.legislation.tn/detailtexte/D%C3%A9cret-num-1996-269-du----jort-1996-017__1996017002693) (accès le 03 12, 2019).
- [10]. s.d. <https://www.supinfo.com/articles/single/2789-monitoring-supervision> (accès le 04 12, 2019).
- [11]. . <https://www.fr.paessler.com/> (accès le 05 01, 2019).
- [12]. s.d. [https://fr.wikipedia.org/wiki/Supervision\\_\(informatique\)](https://fr.wikipedia.org/wiki/Supervision_(informatique)), <https://www.supinfo.com/articles/single/2789-monitoring-supervision>.
- [13]. s.d. <https://fr.wikipedia.org/wiki/Zabbix> (accès le 05 03, 2019).
- [14]. s.d. <https://www.supinfo.com/articles/single/2031-shinken> (accès le 04 12, 2019).
- [15]. s.d. <https://www.monitoring-fr.org/solutions/centreon/> (accès le 03 2019).
- [16]. s.d. [https://fr.wikipedia.org/wiki/Shinken\\_\(logiciel\)](https://fr.wikipedia.org/wiki/Shinken_(logiciel)) (accès le 02 10, 2019).
- [17]. s.d. <https://www.youtube.com/watch?v=nO5jgf8nNjo> (accès le 03 02, 2019).
- [18]. s.d. <https://www.youtube.com/watch?v=nO5jgf8nNjo> (accès le 03 2019).
- [19]. s.d. <https://wiki.monitoring-fr.org/shinken/shinken-work> (accès le Mars 2019).
- [2]. s.d. [https://www.memoireonline.com/04/12/5604/m\\_Monitoring-dune-infrastructure-informatique-sur-base-doutils-libres14.html](https://www.memoireonline.com/04/12/5604/m_Monitoring-dune-infrastructure-informatique-sur-base-doutils-libres14.html).
- [20]. s.d. [https://www.google.fr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjmuNy2s\\_XhAhUoAWMBHThKCbKQjRx6BAgBEAQ&url=http%3A%2F%2Fwww.mathrice.org%2FIMG%2Fpdf%2FslidesShinken.pdf&psig=AOvVaw2jdLtFR3JBmcFf2w2nneq2&ust=1556630681824733](https://www.google.fr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwjmuNy2s_XhAhUoAWMBHThKCbKQjRx6BAgBEAQ&url=http%3A%2F%2Fwww.mathrice.org%2FIMG%2Fpdf%2FslidesShinken.pdf&psig=AOvVaw2jdLtFR3JBmcFf2w2nneq2&ust=1556630681824733) (accès le 03 2019).
- [21]. s.d. <https://www.google.fr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&ved=2ahUKEwj7kM7FIKLiAhXi8eAKHX79AkMQjRx6BAgBEAQ&url=http%3A%2F%2Fwww.mathrice.org%2FIMG%2Fpdf%2FslidesShinken.pdf&psig=AOvVaw3D1naaz5mZu8zKxTn0Ufyj&ust=1558168295835254> (accès le Avril 2019).
- [22]. s.d. <https://connect.ed-diamond.com/GNU-Linux-Magazine/GLMFHS-049/Shinken-quand-un-Python-rencontre-Nagios> (accès le 04 05, 2019).
- [23]. s.d. [https://fr.wikipedia.org/wiki/UML\\_\(informatique\)](https://fr.wikipedia.org/wiki/UML_(informatique)) (accès le Mais 2019).
- [3]. s.d. <https://www.appvizer.fr/magazine/services-informatiques/supervision-info/supervision-informatique> (accès le 02 2019).
- [4]. s.d. [https://fr.wikipedia.org/wiki/Intelligent\\_Platform\\_Management\\_Interface](https://fr.wikipedia.org/wiki/Intelligent_Platform_Management_Interface).
- [5]. s.d. [www.lipn.univ-paris13.fr/~kanawati/m3108/supervision-RCPW01.pdf](http://www.lipn.univ-paris13.fr/~kanawati/m3108/supervision-RCPW01.pdf) (accès le 03 2019).
- [6]. s.d. <http://supervision.clever.fr/protocole-snmp-simple-network-management-protocole/> (accès le 04 2019).
- [7]. s.d. <https://www.supinfo.com/articles/single/1639-protocole-snmp>.
- [8]. 15 Mai . <https://www.slideshare.net/anshumanbiswal/snmp-15866756>.
- [9]. s.d. [https://www.memoireonline.com/05/10/3511/m\\_Implementation-dune-application-de-gestion-de-reseau-base-sur-le-protocole-SNMP8.html](https://www.memoireonline.com/05/10/3511/m_Implementation-dune-application-de-gestion-de-reseau-base-sur-le-protocole-SNMP8.html) (accès le Mai 16, 2019).

# Annexe A

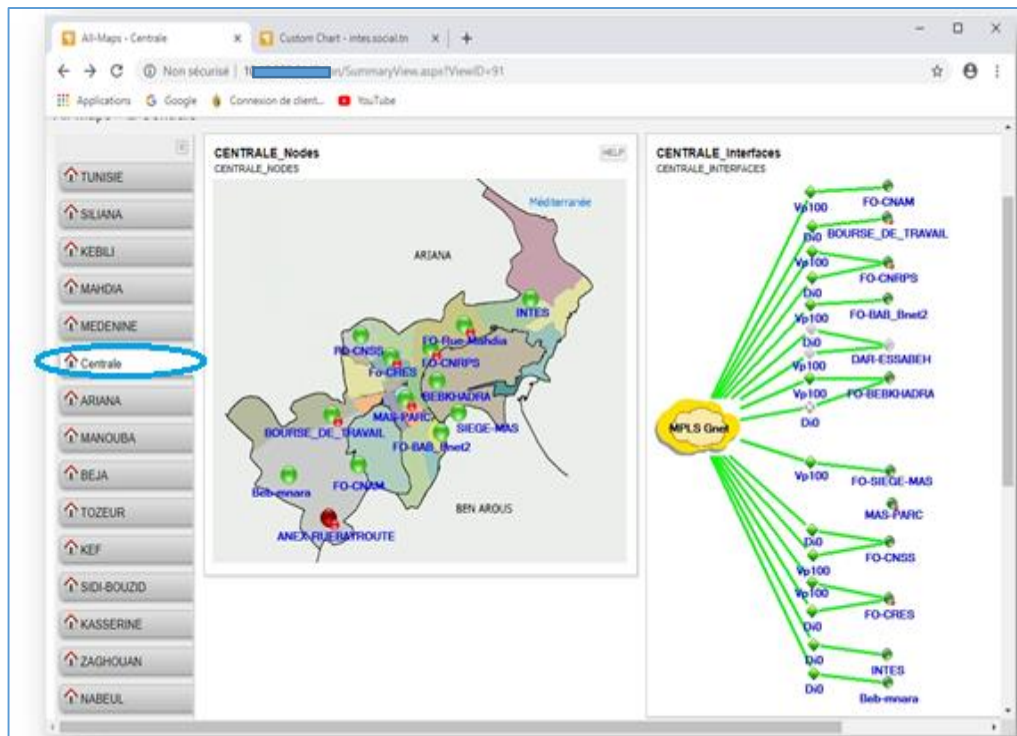
## SolarWinds

Maps, de la Tunisie. Tous les gouvernorats sont interconnectés à la centrale.



Maps de la centrale

Le schéma suivant montre, l'état des routeurs de la centrale.



# Maps INTES : Institut National de Travail des Etudes Sociales

Node Details - Summary - intes.social.tn - Summary

**Management** [HELP]

**Node Details** [HELP]

Node Status ● Node is Up.

Polling IP Address [redacted]

Dynamic IP No

Machine Type Cisco

DNS

System Name intes.social.tn

Description Cisco IOS Software, C860 Software (C860VAE-ADVSECURITYK9\_NPE-M), Version 15.2(4)M4, RELEASE SOFTWARE (fc2) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2013 by Cisco Systems, Inc. Compiled Thu 20-Jun-13 16:36 by prod\_rel\_team

Location

Contact

SysObjectID 1.3.6.1.4.1.9.1.1358

Last Boot mercredi 20 mars 2019 13:56

Operating System 15.2(4)M4, RELEASE SOFTWARE (fc2)

IOS Image C860VAE-ADVSECURITYK9\_NPE-M

Hardware Physical

No of CPUs 1

Telnet telnet:[redacted]:4

Web Browse http:[redacted]

**Active Alerts on This Node**  
ALL UNACKNOWLEDGED ALERTS

TIME OF ALERT NETWORK DEVICE

**Availability Statistics**

PERIOD

Today

Yesterday

Last 7 Days

Last 30 Days

This Month

Last Month

This Year

**Event Summary**

**Top CPUs by Percent Load**

May 23 20

Zoom 1h 12h 24h

# Maps INTES : Institut National de Travail des Etudes Sociales

Node Details - Summary - intes.social.tn - Summary

**Management** [HELP]

**Node Details** [HELP]

Node Status ● Node is Up.

Polling IP Address [redacted]

Dynamic IP No

Machine Type Cisco

DNS

System Name intes.social.tn

Description Cisco IOS Software, C860 Software (C860VAE-ADVSECURITYK9\_NPE-M), Version 15.2(4)M4, RELEASE SOFTWARE (fc2) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2013 by Cisco Systems, Inc. Compiled Thu 20-Jun-13 16:36 by prod\_rel\_team

Location

Contact

SysObjectID 1.3.6.1.4.1.9.1.1358

Last Boot mercredi 20 mars 2019 13:56

Operating System 15.2(4)M4, RELEASE SOFTWARE (fc2)

IOS Image C860VAE-ADVSECURITYK9\_NPE-M

Hardware Physical

No of CPUs 1

Telnet telnet:[redacted]:4

Web Browse http:[redacted]

**Active Alerts on This Node**  
ALL UNACKNOWLEDGED ALERTS

TIME OF ALERT NETWORK DEVICE

**Availability Statistics**

PERIOD

Today

Yesterday

Last 7 Days

Last 30 Days

This Month

Last Month

This Year

**Event Summary**

**Top CPUs by Percent Load**

May 23 20

Zoom 1h 12h 24h



# Maps BEBKHADRA

**Node Details - Network - FO-BEBKHADRA.social.tn - Network**

**Hardware Details** (HELP)

Hardware Status: Up

Manufacturer: Cisco

Model: CISCO1921/K9

Service Tag: FCZ1817C3BL

Last Poll Time: 23/05/2019 09:02:48

**EnergyWise Node Details** (HELP)

Management: Manage EnergyWise

Status: EnergyWise is Disabled

**Routing Neighbors** (0 records) (HELP)

NODE NAME	PROTOCOL	STATUS	IP ADDRESS	REMOTE AS	LAST CHANGE
[Empty Table]					

**Routing Table** (17 records) (HELP)

DESTINATION NETWORK	CIDR	NEXT HOP	INTERFACE	METRIC	SOURCE
0.0.0.0	0		InterfaceIndex #1	0	NetMgmt

**Top 10 Flapping Routes**

Main | Last 7 days

DESTINATION NETWORK	CIDR	FLAPS	NEXT
[Empty Table]			

**Default Route Changes** (0 records)

Main | Last 7 days

NEXT HOP	ROUTE CHANGE
[Empty Table]	

**Current Percent Utilization of Each Interface**

STATUS	INTERFACE
Up	Virtual-PPP100 - Vp100
Shutdown	Dialer0 - Di0

**Hardware Health**

Power Supply (V) | Temperature

**FO-BEBKHADRA.s**  
May 23 2019, 12:00 am - May 2

## Annexe B

### WinsCP et Putty

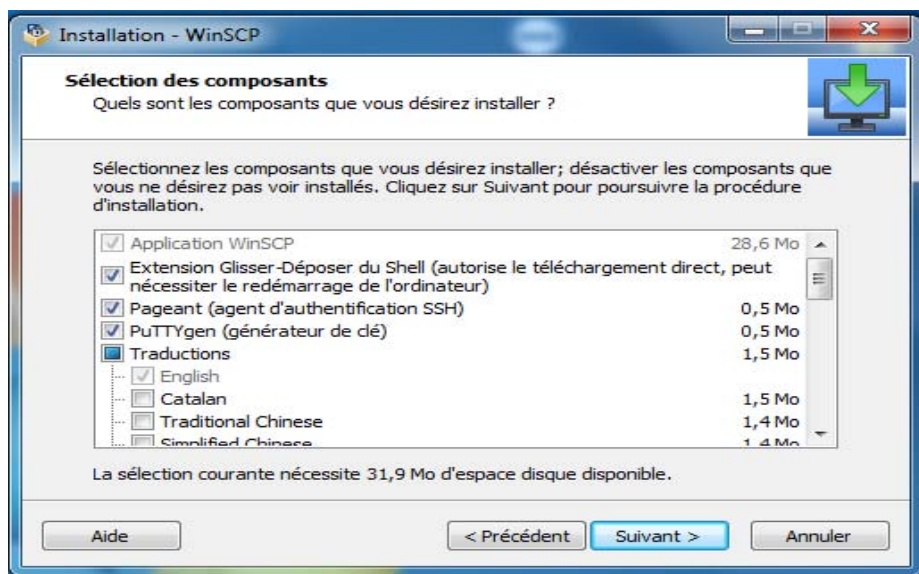
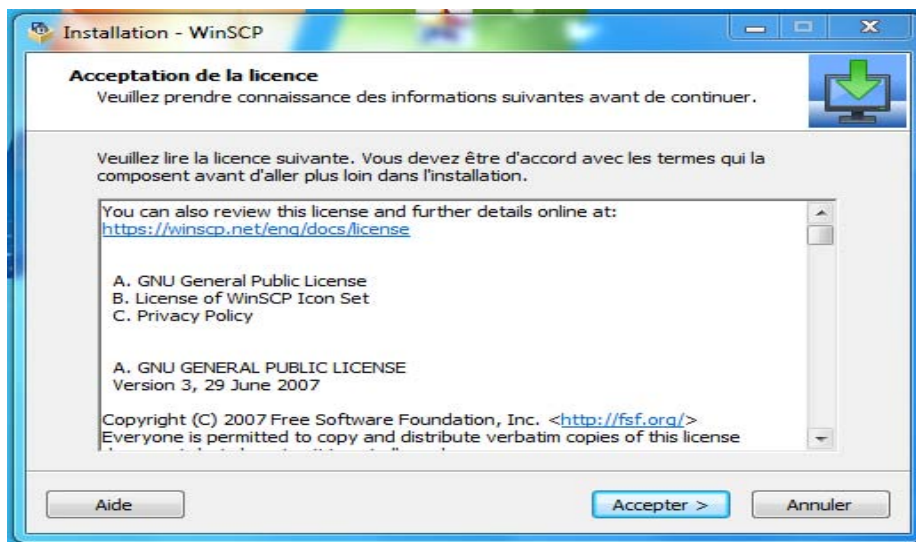
#### 1. WinsCP

##### Définition

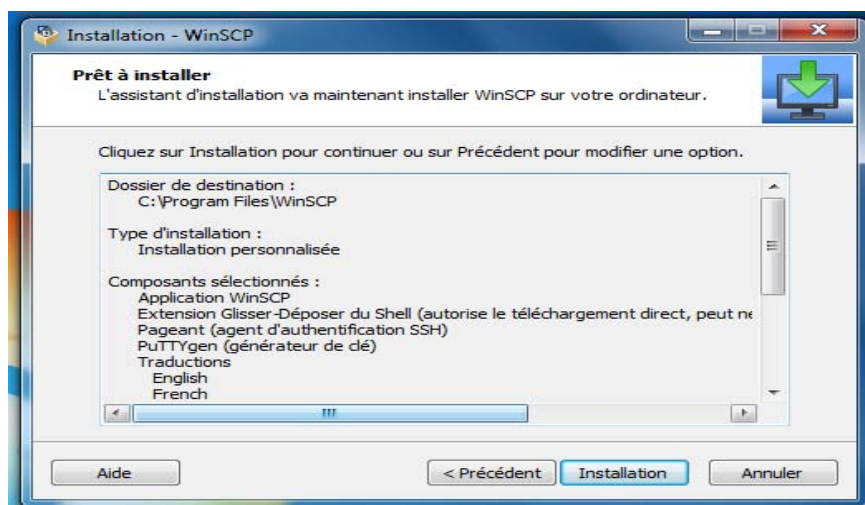
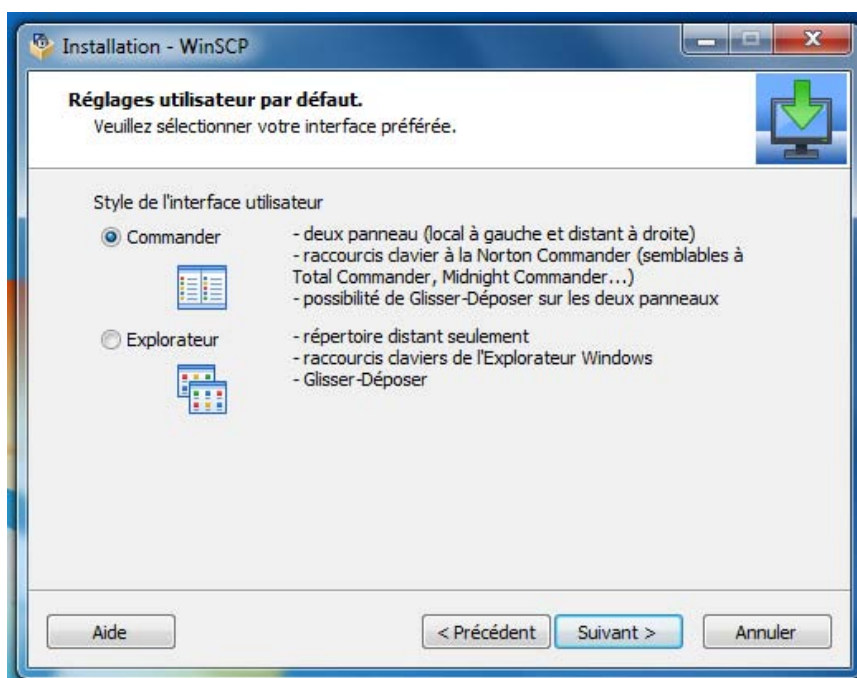
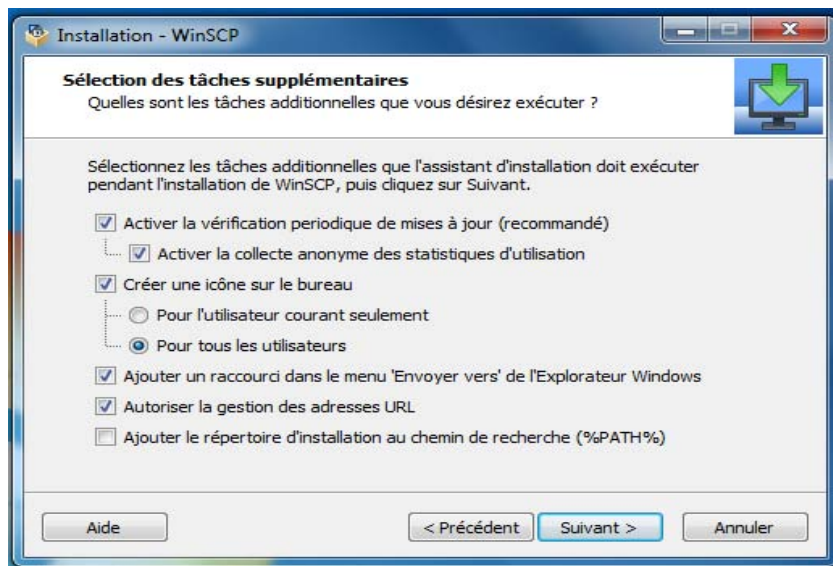
WinsCP est un client SFTP (Secure FTP) graphique pour Windows. Il utilise SSH (Secure Shell) et il est open source. Le but de ce programme est de permettre la copie sécurisée de fichiers entre un ordinateur local et un ordinateur distant.

##### Installation

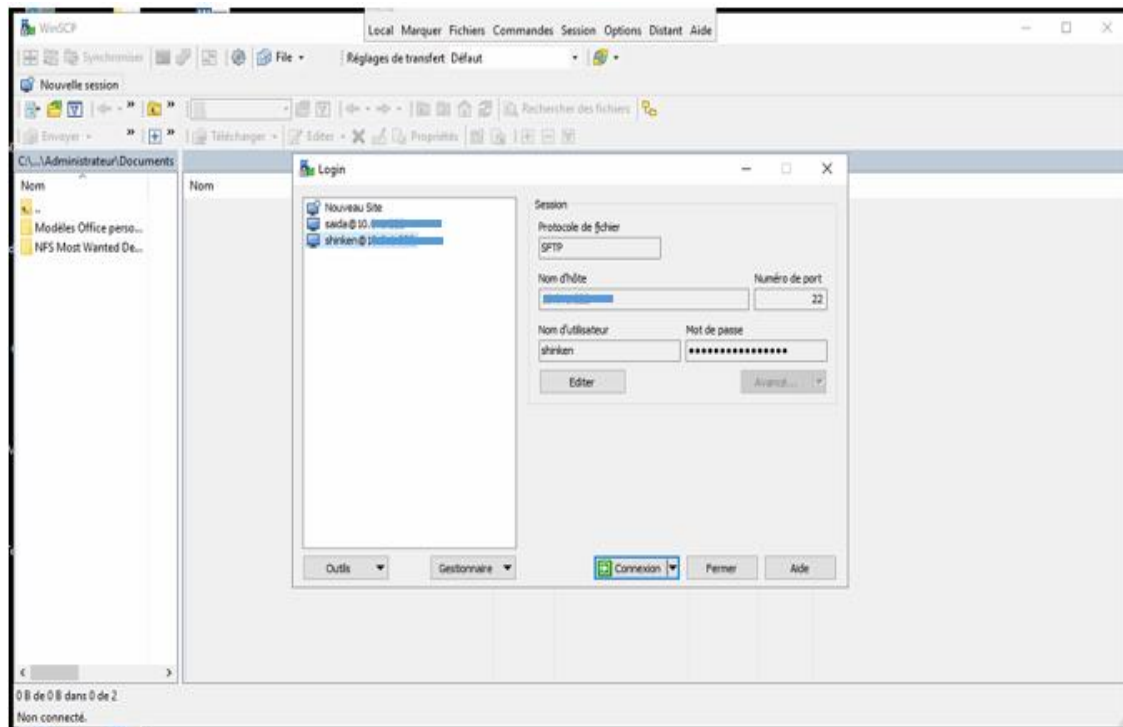
Téléchargez l'application à l'adresse suivante : <http://winscp.net/>



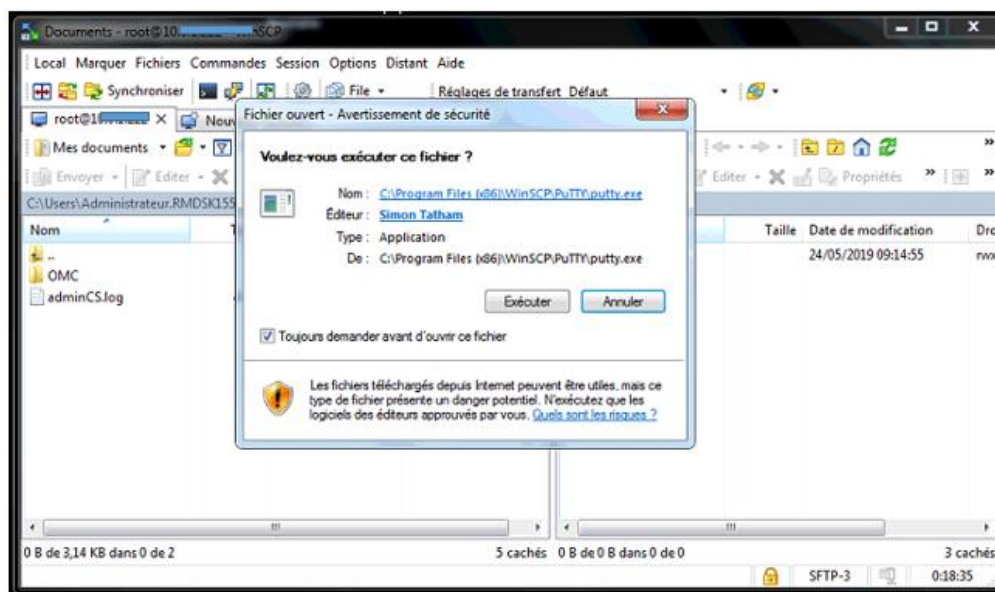




Pour la première fois on obtiendra cette fenêtre, il ne faut pas indiquer le mot de passe à ce moment pour des raisons de sécurité. Il sera demandé au moment de connexion.



Ouvrir une session Putty



```
root@1
Using username "root".
Linux Saïda-Shinken 4.9.0-3-amd64 #1 SMP Debian 4.9.30-2+deb9u5 (2017-09-19)
_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri May 24 15:50:22 2019 from 1[redacted]
root@Saïda-Shinken:~#
```

## 2. Putty

Putty, est un outil permettant la connexion à distance. Il est disponible pour Windows, diverses plates-formes Unix (et non-officiellement sur d'autres plates-formes).

**PuTTY**

Auteur/éditeur : **Simon Tatham**

Présentation [Telecharger.com](#) Avis des utilisateurs Captures d'écran

**Note des utilisateurs :**  
 ★★★★★ 2 avis  
 [Donnez une note]

téléchargé les 7 derniers jours  
**1368 fois**

**↓ télécharger**  
 gratuitement

**PuTTY** est un programme permettant de se connecter à distance à des serveurs en utilisant les protocoles SSH, Telnet ou Rlogin. L'ensemble des sessions peuvent être automatiquement enregistrées dans un rapport qui pourra être consulté ultérieurement. La fenêtre de commandes est personnalisable afin de convenir à tous les utilisateurs : il est possible de modifier le type de curseur, les couleurs, la police de caractère utilisée, etc. Les connexions sont également paramétrables : il est possible de passer par un proxy, de préférer une connexion SSH 1 ou SSH 2, de mettre en place la compression lors des sessions SSH, d'utiliser un mode passif pour les négociations en Telnet, etc.

<b>Version :</b>	0.67
<b>Licence :</b>	Gratuit
<b>Taille :</b>	0.51 Mo
<b>Configuration minimale :</b>	Windows XP/Vista /7/8/8.1/10
<b>Date de sortie :</b>	29/02/2016
<b>Langue :</b>	Anglais

- Vous avez un problème avec ce logiciel, consultez les forums
- Les logiciels les mieux notés de cette catégorie
- Tout savoir sur le téléchargement avec 01net

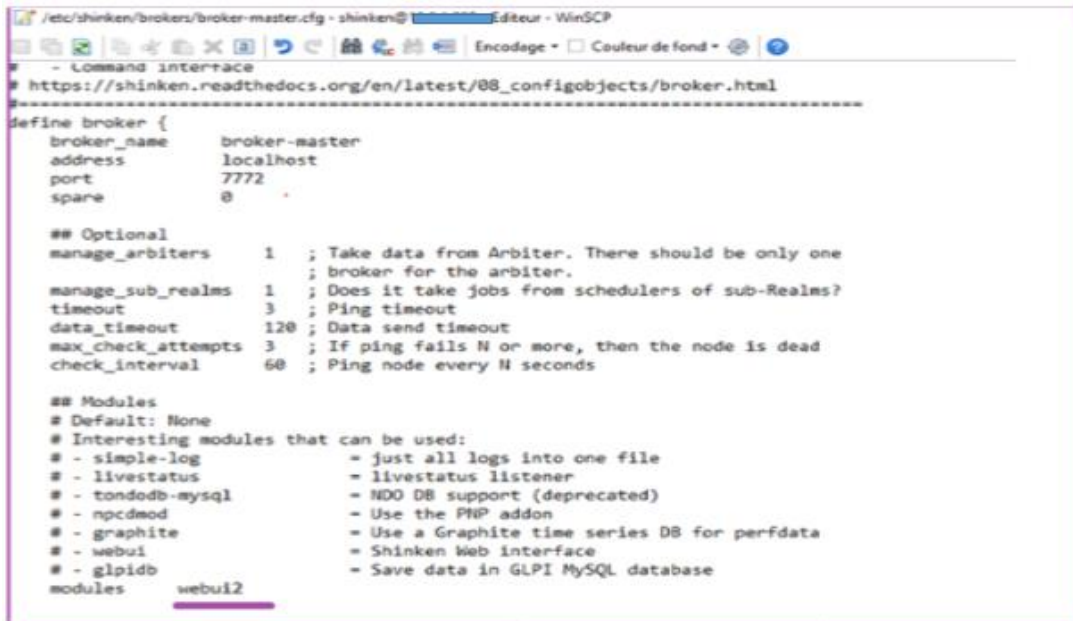
## Annexe C

### La configuration du serveur shinken

#### 1. Configuration de l'interface webui2

La configuration de l'interface se fait dans le fichier `/etc/shinken/brokers/broker-master.cfg`

Editer le fichier pour activer le module `webui2`



```
/etc/shinken/brokers/broker-master.cfg - shinken@ - Editeur - WinSCP
- L'ommande interface
# https://shinken.readthedocs.org/en/latest/08_configobjects/broker.html
-----
define broker {
    broker_name      broker-master
    address          localhost
    port             7772
    spare            0

    ## Optional
    manage_arbiters  1 ; Take data from Arbiter. There should be only one
                    ; broker for the arbiter.
    manage_sub_realmes 1 ; Does it take jobs from schedulers of sub-Realms?
    timeout          3 ; Ping timeout
    data_timeout     120 ; Data send timeout
    max_check_attempts 3 ; If ping fails N or more, then the node is dead
    check_interval   60 ; Ping node every N seconds

    ## Modules
    # Default: None
    # Interesting modules that can be used:
    # - simple-log          = just all logs into one file
    # - livestatus         = livestatus listener
    # - tondodb-mysql      = NDO DB support (deprecated)
    # - npcdmod            = Use the PHP addon
    # - graphite           = Use a Graphite time series DB for perfdata
    # - webui              = Shinken Web Interface
    # - glpidb             = Save data in GLPI MySQL database
    modules              webui2
```

#### 2. Démarrage des services

On démarre maintenant les services pour prendre en compte la nouvelle configuration.  
On relance shinken `/etc/init.d/shinken start`

#### 3. Authentification et changement de mot de passe par défaut

On peut changer le mot de passe admin par défaut, cela par la modification du fichier `/etc/shinken/contacts/admin.cfg`, On se connecte à l'interface web à l'aide des identifiants administrateur qui sont par défaut : admin / admin

```

/etc/shinken/contacts
Nom
  admin.cfg
  guest.cfg
  /etc/shinken/contacts/admin.cfg - shinken@ - Editeur - WinSCP
  # This is a default admin
  # CHANGE ITS PASSWORD!

  define contact{
    use                generic-contact
    contact_name       admin
    password            admin
    #host_notifications_enabled 1
    #service_notifications_enabled 1
    #service_notification_period 24x7
    #host_notification_period 24x7
    #service_notification_options c,w,r
    #host_notification_options d,u,r,f,s
    service_notification_commands notify-service-by-email
    email              saida.intes@gmail.com
    # notification ways email_in_day
    can_submit_commands 1
  }

```

#### 4. Ajouter un host

Maintenant, pour ajouter une hôte, il suffit de créer un fichier '. cfg' par hôte dans le dossier /etc/shinken/hosts/. Voilà le fichier de configuration de notre serveur shinken

```

/etc/shinken/hosts
Nom
  localhost
  R-Beb-Bnet-1
  R-Beb-Bnet-2
  R-Beb-Bnet-3
  R-Beb-Bnet-4
  R-Beb-Bnet-5
  R-Beb-Bnet-6
  R-Beb-Bnet-7
  R-Beb-Bnet-8
  R-Beb-Bnet-9
  R-Beb-Bnet-10
  /etc/shinken/hosts/localhost.cfg - shinken@ - Editeur - WinSCP
  define host{
    use                linux-snmp
    host_name          serveur-shinken
    address            192.168.1.100
    check_interval     5
    retry_interval     1
    max_check_attempts 5
    _SNMPCOMMUNITY    public
  }

```

Fichier de configuration du routeur Beb-Bnet1

```

/etc/shinken/hosts
Nom
  localhost.cfg
  R-Beb-Bnet-1
  /etc/shinken/hosts/R-Beb-Bnet-1.cfg - shinken@ - Editeur - WinSCP
  define host{
    use                cisco,routen
    host_name          R-Beb-Bnet-1
    address            192.168.1.100
    check_interval     5
    retry_interval     1
    max_check_attempts 5
    _SNMPCOMMUNITY    public
  }

```

#### 5. Ajout des plugins snmp et Nagios

Il faut installer les différents paquets (en utilisateur root) :



**Snmpd** : Permet de rendre accessible une machine « A » dont on souhaite récupérer les informations de fonctionnement pour les exploiter sur une autre machine « B »

```
apt-get install snmp
```

```
source liste (ajouter contrib non-free) (/etc/apt/source.list)
```

```
apt-get update
```

```
apt-get install snmp-mibs-downloader
```

```
(vim /etc/default/snmpd ==> export MIBS=/usr/share/mibs)
```

```
/etc/init.d/snmpd restart
```

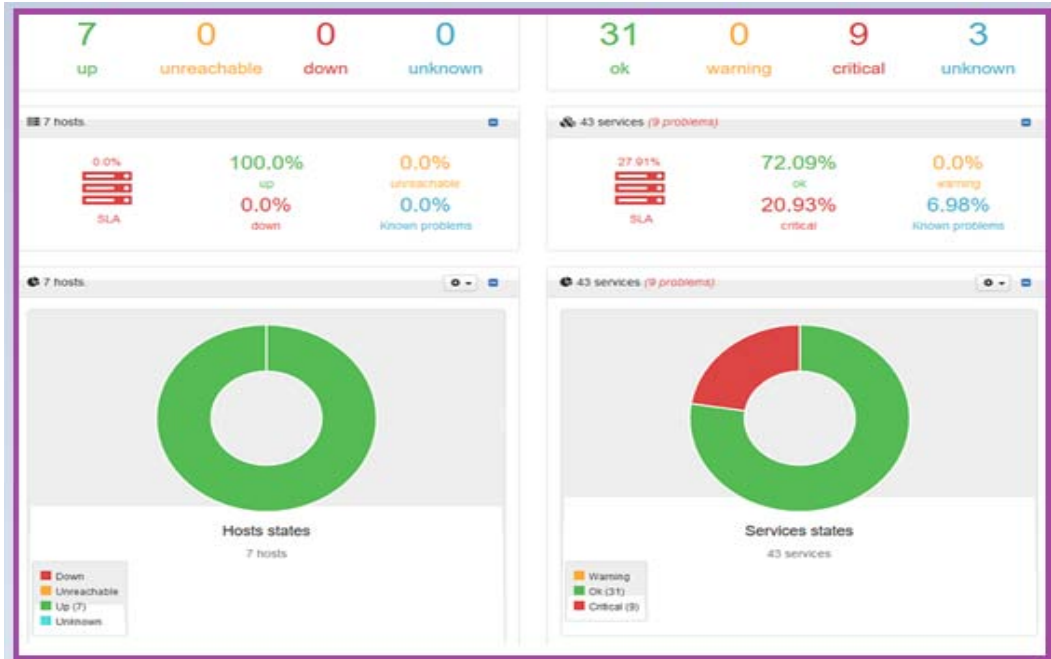
**ntp** : Est un protocole qui permet de synchroniser via un réseau informatique, l'horloge locale d'ordinateurs sur une référence d'heure

**Nagios-plugins** : Sont les plugins de nagios qui vont permettre la remontée d'information (les plugins sont compatibles avec Shinken).

## Annexe D

### Fonctionnement du serveur shinken

Les sept hosts sont UP,



7 hosts

Business impact: Normal

Host	Service	State	Duration	Output
<input type="checkbox"/> R-Beb-Bnet-1		UP	18h 1m	PING OK - Packet loss = 0%, RTA = 0.56 ms
<input type="checkbox"/> R-Beb-Bnet-2		UP	18h 1m	PING OK - Packet loss = 0%, RTA = 0.81 ms
<input type="checkbox"/> R-Beb-Khadhra		UP	18h 2m	PING OK - Packet loss = 0%, RTA = 2.70 ms
<input type="checkbox"/> R-Dar-Assabah		UP	17h 59m	PING OK - Packet loss = 0%, RTA = 2.14 ms
<input type="checkbox"/> R-Parc-Auto		UP	17h 58m	PING OK - Packet loss = 0%, RTA = 24.49 ms
<input type="checkbox"/> R-Rue-Beyrouf		UP	17h 59m	PING OK - Packet loss = 0%, RTA = 194.74 ms
<input type="checkbox"/> serveur-shinken		UP	18h 1m	OK - 0.005ms, lost 0%

Les services sont OK

- ✓ CPU
- ✓ Hardware
- ✓ InterfaceUsage
- ✓ Memory

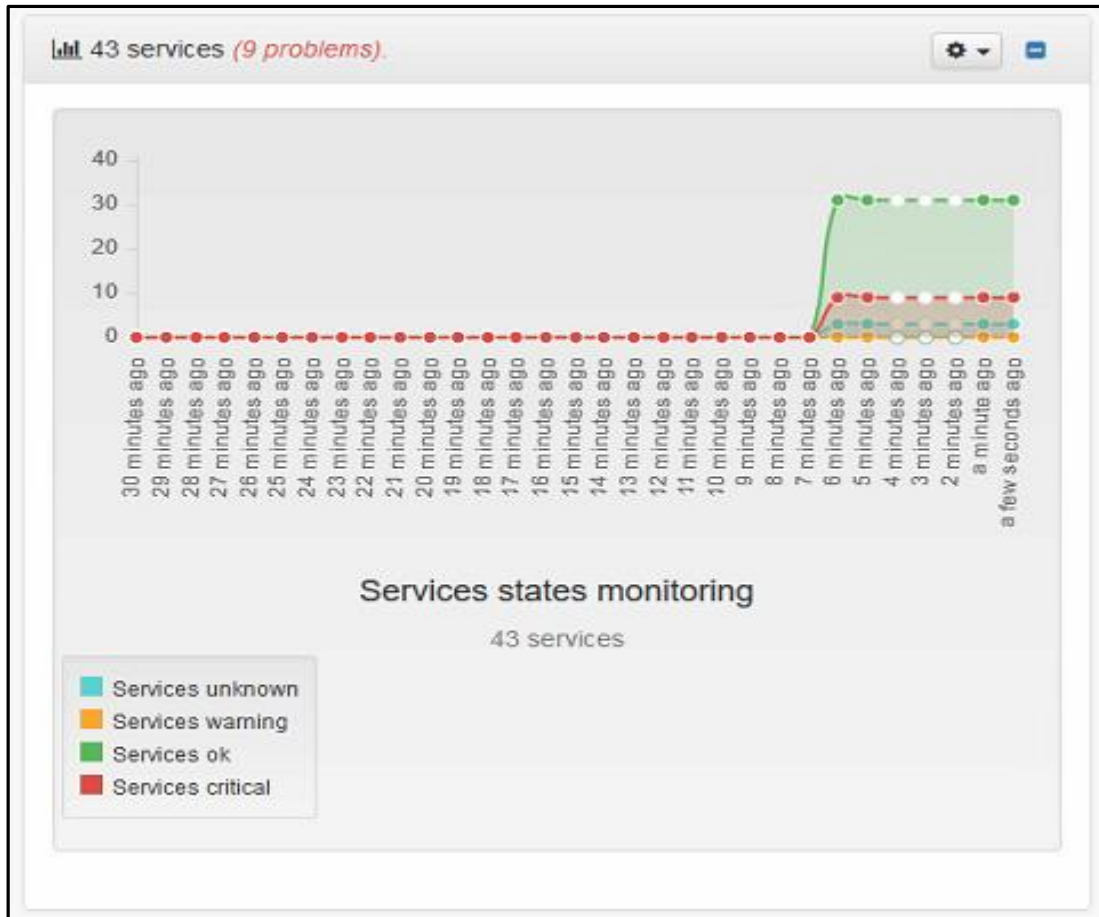
<input type="checkbox"/>		R-Beb-Bnet-1	Cpu	OK	3h 27m	OK - cpu 1 usage (5 min avg.) is 2.00%
<input type="checkbox"/>			Hardware	OK	3h 25m	OK - environmental hardware working fine
<input type="checkbox"/>			InterfaceErrors	OK	3h 27m	OK - interface Embedded-Service-Engine0/0 errors I...
<input type="checkbox"/>			InterfaceUsage	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 usage i...
<input type="checkbox"/>			Memory	OK	3h 24m	OK - mempool Processor_1 usage is 14.33%, mempo...
<input type="checkbox"/>		R-Beb-Bnet-2	Cpu	OK	3h 22m	OK - cpu 1 usage (5 min avg.) is 1.00%
<input type="checkbox"/>			Hardware	OK	3h 25m	OK - environmental hardware working fine
<input type="checkbox"/>			InterfaceErrors	OK	3h 26m	OK - interface Embedded-Service-Engine0/0 errors I...
<input type="checkbox"/>			InterfaceUsage	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 usage i...
<input type="checkbox"/>			Memory	OK	3h 24m	OK - mempool Processor_1 usage is 14.34%, mempo...
<input type="checkbox"/>		R-Beb-Bnet-3	Cpu	OK	3h 22m	OK - cpu 1 usage (5 min avg.) is 1.00%

<input type="checkbox"/>		R-Dar-Assabah	Cpu	OK	3h 28m	OK - cpu 1 usage (5 min avg.) is 0.00%
<input type="checkbox"/>			Hardware	OK	3h 27m	OK - environmental hardware working fine
<input type="checkbox"/>			InterfaceErrors	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 errors I...
<input type="checkbox"/>			InterfaceUsage	OK	3h 24m	OK - interface Embedded-Service-Engine0/0 usage i...
<input type="checkbox"/>			Memory	OK	3h 28m	OK - mempool Processor_1 usage is 9.30%, mempo...
<input type="checkbox"/>		R-Parc-Auto	Cpu	OK	3h 24m	OK - cpu 1 usage (5 min avg.) is 4.00%
<input type="checkbox"/>			Hardware	OK	3h 27m	OK - environmental hardware working fine, environm...
<input type="checkbox"/>			InterfaceErrors	OK	3h 26m	OK - interface ATM0 (alias AT0) errors in 0.00/s out 0...
<input type="checkbox"/>			InterfaceUsage	OK	3h 29m	OK - interface ATM0 (alias AT0) usage is in 0.00% (0...
<input type="checkbox"/>			Memory	OK	3h 28m	OK - mempool Processor usage is 33.56%, mempool ...
<input type="checkbox"/>		R-Rue-Beyrou	Cpu	OK	3h 26m	OK - cpu 1 usage (5 min avg.) is 3.00%
<input type="checkbox"/>			Hardware	OK	3h 27m	OK - environmental hardware working fine, environm...
<input type="checkbox"/>			InterfaceErrors	OK	3h 24m	OK - interface ATM0 (alias AT0) errors in 0.00/s out 0...
<input type="checkbox"/>			Memory	OK	3h 28m	OK - mempool Processor usage is 26.64%, mempool ...
<input type="checkbox"/>		serveur-shinken	TimeSync	OK	3h 27m	NTP OK. Offset -3.069639206e-06 secs

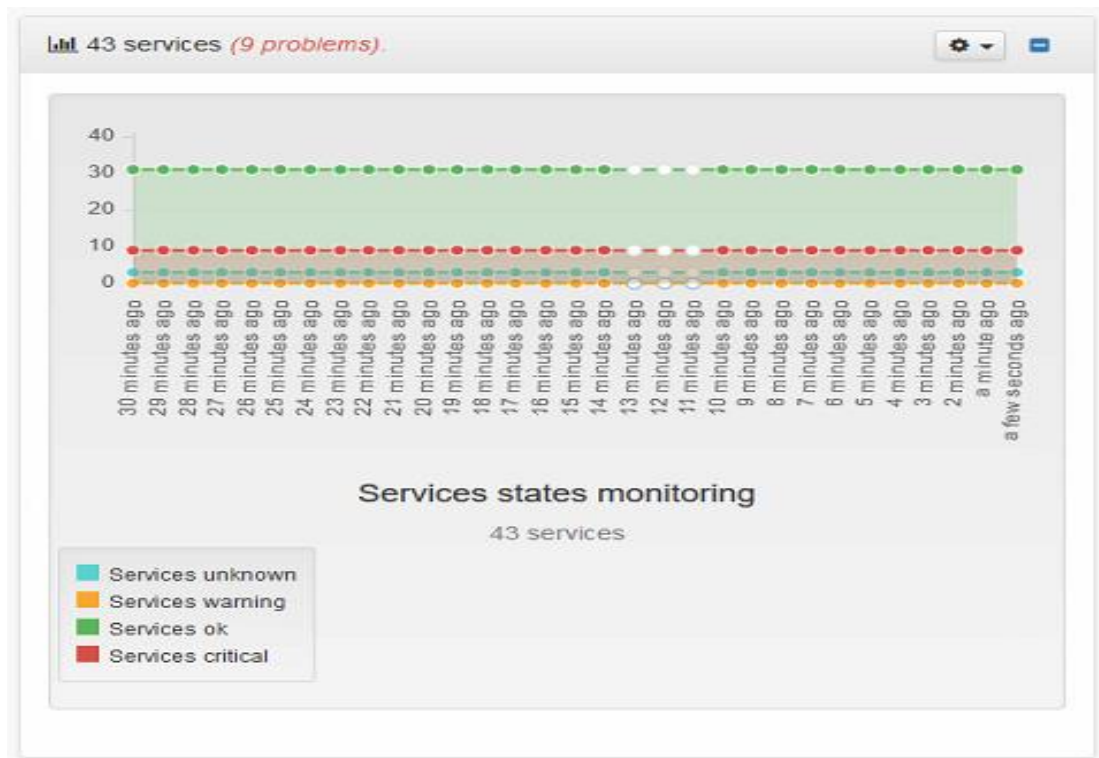
### L'état des services graphiquement

- ✓ 31 OK
- ✓ 9 Critical
- ✓ 3 Unknown
- ✓ 0 Warning



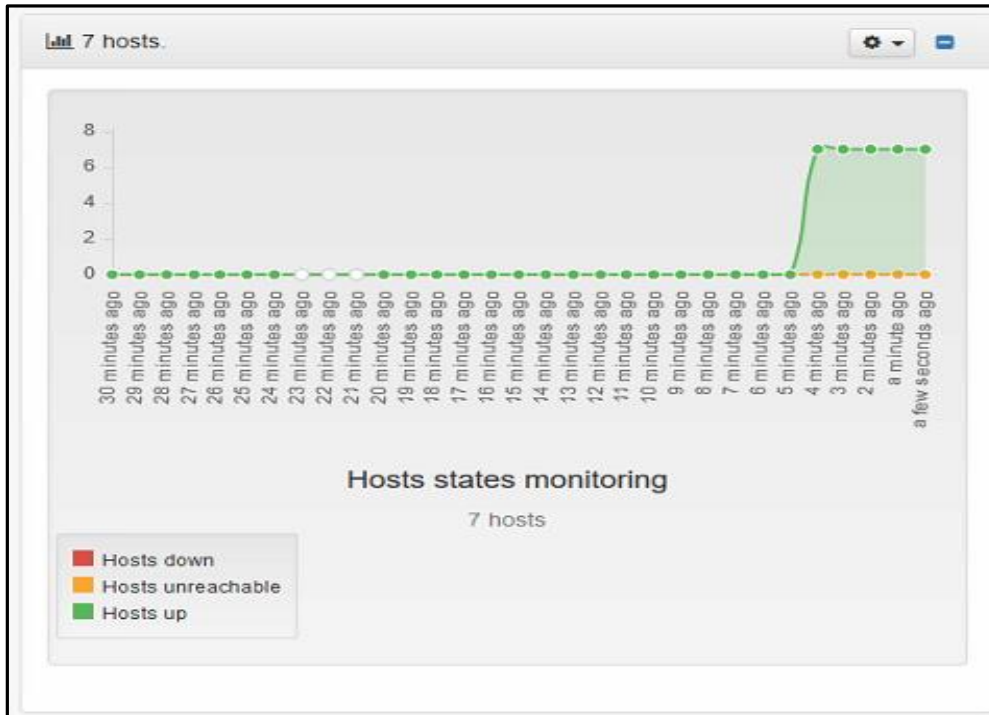


L'état des services graphiquement après 30mn

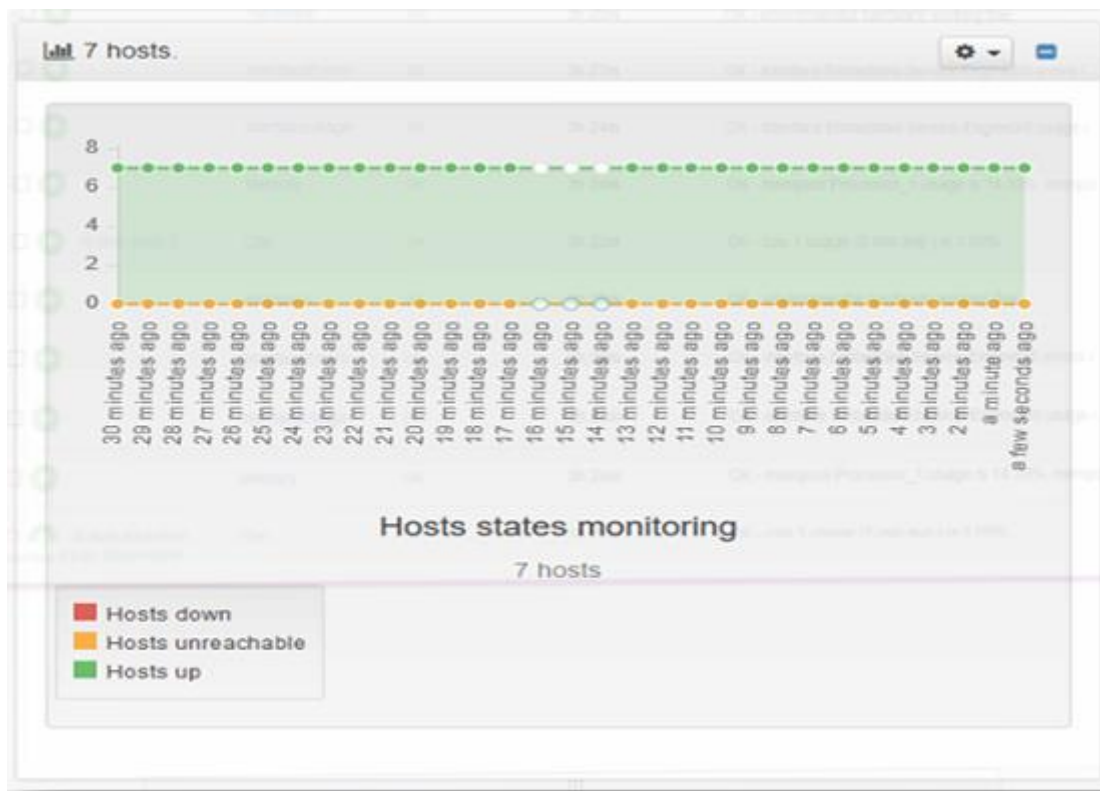


L'état des host graphiquement cinq minutes avant

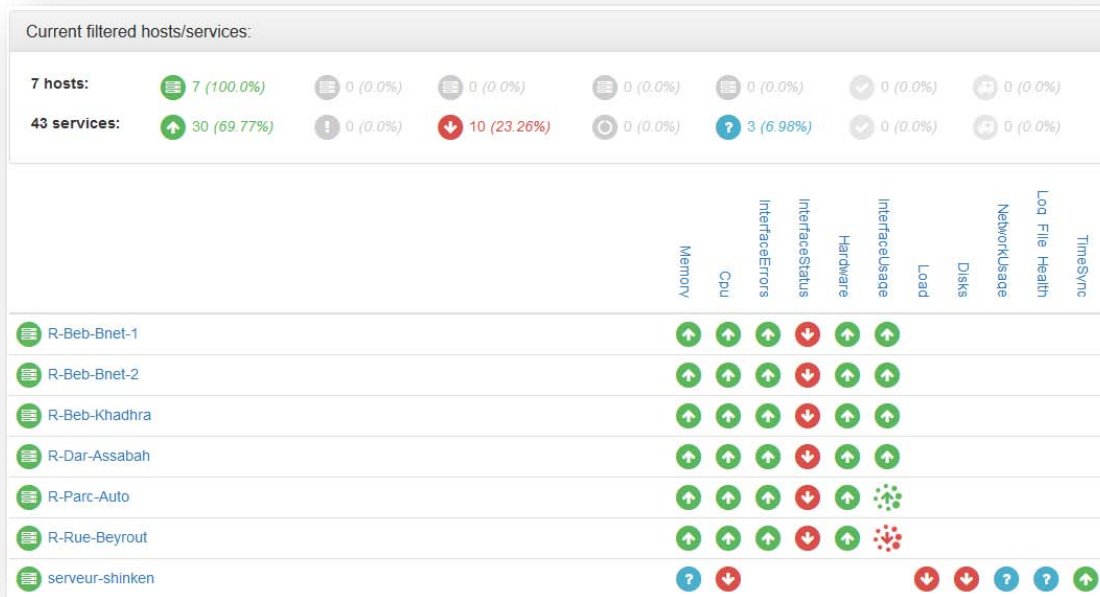
- ✓ 7 host UP
- ✓ 0 host down
- ✓ 0 host unreachable



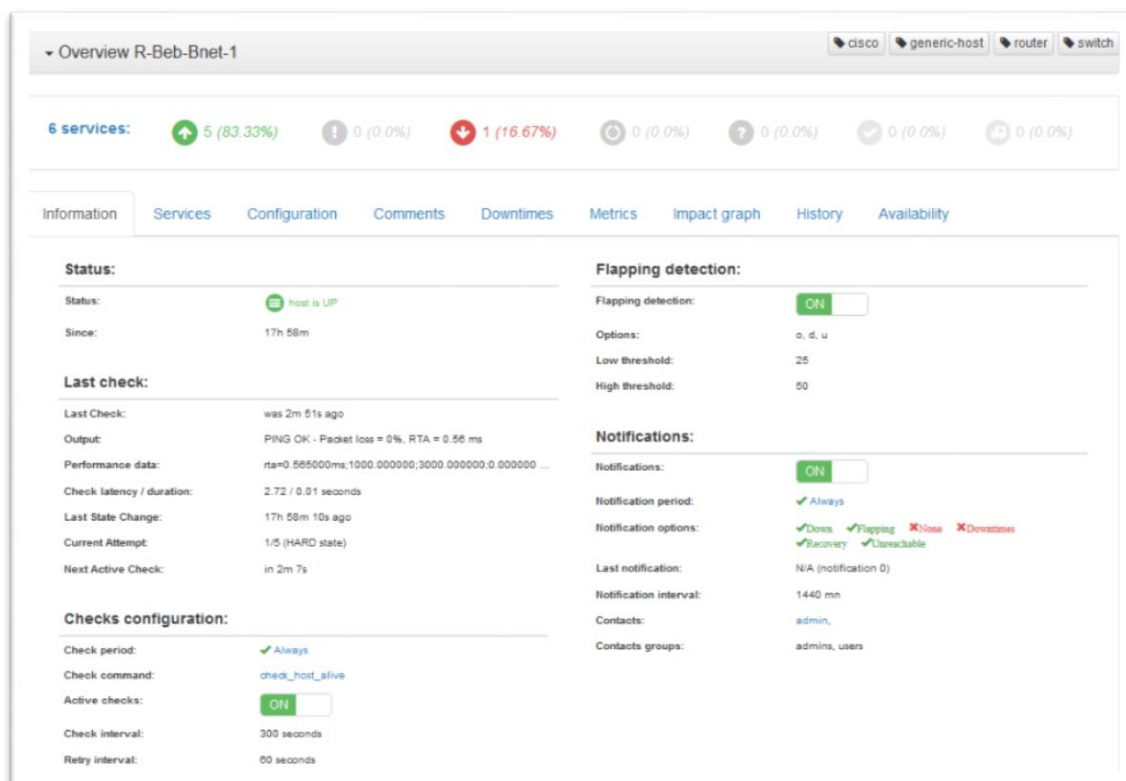
L'état des host graphiquement après 30 minutes



Les services de tous les hosts



### Information des services du routeur Beb-Bnet-1



6 services: ▲ 5 (83.33%) ! 0 (0.0%) ▼ 1 (16.67%) ⏸ 0 (0.0%) ? 0 (0.0%) ✓ 0 (0.0%) ↺ 0 (0.0%)

Information Services Configuration Comments Downtimes Metrics Impact graph History Availability

Service	Metric	Value	Warning	Critical	Min	Max	UOM
Host check	pl	0	100	100	0	100	%
	rta	0.67	1000	3000	0		ms
Cpu	cpu_1_usage	17	80	90	0	100	%
Hardware	mvolt_1	12213	8990	13970			
	mvolt_2	5159	4370	5920			
	mvolt_3	3322	2800	3790			
	mvolt_4	2477	2120	2870			
	mvolt_5	1803	1530	2070			
	mvolt_6	1199	1020	1380			
	mvolt_7	1025	850	1210			

### Etat du routeur Beb-Bnet-1 et ses services

