



Rapport d'apprentissage Master 1 -  
Informatique  
*Supervision open-source d'un système d'information*

Thomas Calatayud

2016-2017

# Table des matières

Remerciements .....	1
Introduction.....	2
Le conseil départemental de Maine-et-Loire .....	3
Présentation du conseil départemental .....	3
L'unité Système de Production et ses missions .....	5
La Supervision .....	6
Présentation de l'outil Nagios.....	7
Etude et mise en place d'une solution libre/open-source de supervision .....	8
Mon activité au sein de l'unité .....	9
Mise à niveau technique .....	9
Découverte, déploiement, installation et configuration de l'outil Nagios .....	12
Synchronisation des nagios de l'unité .....	16
SAEIR, installation des baies de stockage et de serveurs. Déplacement dans les salles. Et installation du nagios. ....	16
Identification des solutions libres/open-sources de supervision .....	17
Instantané des solutions libres/open-sources. ....	17
Etablir les critères de sélection.....	18
Présentation des solutions sélectionnées.....	19
Solution 1.....	19
Solution 2.....	19
Bilan.....	19
Annexes .....	20
Table des illustrations .....	21
Sources .....	22

---

Rapport rédigé par Thomas CALATAYUD  
Étudiant en Master Informatique à l'université d'Angers

Responsable de stage  
M. Frédéric LARDEUX  
Enseignant chercheur  
LERIA, Université d'Angers  
[frederic.lardeux@univ-angers.fr](mailto:frederic.lardeux@univ-angers.fr)

Tuteur en entreprise  
M. Denis PITHON  
Responsable de l'unité système de production  
Département de Maine et Loire  
[d.pithon@maine-et-loire.fr](mailto:d.pithon@maine-et-loire.fr)

## Remerciements

---

# Introduction

En sortant de ma License Informatique, je souhaitais continuer mon parcours vers un Master Informatique. Deux choix se sont offert à moi : continuer mes études dans un parcours orienté vers la recherche ou dans un parcours professionnalisant.

Mon premier choix était particulièrement porté vers le parcours recherche. Je me suis donc inscrit en première année de Master Informatique qui est générale aux deux parcours, dans l'optique de poursuivre sur une deuxième année en Master Intelligence Décisionnelle.

Il m'a été rappelé qu'il était notamment possible d'effectuer ces deux années de Master en alternance en entreprise.

Malgrès ma passion pour les études, apprendre de nouvelles choses, la découverte et la recherche de nouveauté, l'idée d'effectuer ce Master en alternance m'a plutôt attiré, j'ai donc décidé de me lancer dans la recherche d'une entreprise, prête à m'accueillir et à m'offrir un sujet d'apprentissage qui puisse m'intéresser.

C'est à ce moment, qu'on m'a proposé une offre au Conseil Départemental de Maine-et-Loire. J'ai donc intégré le service Système de Production au sein de la Direction Logistique et Système d'Information en ayant pour projet de remettre à neuf le système de supervision. Pour cela on m'a demandé d'identifier et de proposer une solution libre et open-source et éventuellement de la mettre en place.

J'ai finalement choisis l'apprentissage pour goûter à l'informatique dans un contexte professionnel. Cela me permet d'avoir un peu plus d'expérience professionnelle dans le domaine informatique. Et grâce à cette alternance je peux continuer d'étudier à l'université tout en apprenant de nouvelles chose au Conseil Départemental. Donc j'ai l'opportunité d'étudier et de travailler en ayant un salaire dans un domaine qui me plait. Et j'ai accepter ce sujet dans le but de découvrir l'administration système, un aspect de l'informatique que je ne connaissait pas beaucoup.

# Le conseil départemental de Maine-et-Loire

## Présentation du conseil départemental

(Voir notes de la réunion d'accueil) Qu'est ce que c'est ? Qu'est ce qu'il fait , Organisation ? (élus, administration territoriale, directions, services ...) - Une collectivité territoriale - Missions/compétences - Organisation

### ORGANIGRAMME

du Département  
de Maine-et-Loire



Mise à jour - Mai 2017

maine-et-loire.fr

DÉPARTEMENT DE MAINE-ET-LOIRE  
**anjou**

Image 1. Organigramme du Conseil Départemental

- La DLSI : qu'est ce qu'on fait ?
- Les différents services
- L'unité système de production

#### Direction Logistique et Systèmes d'Information

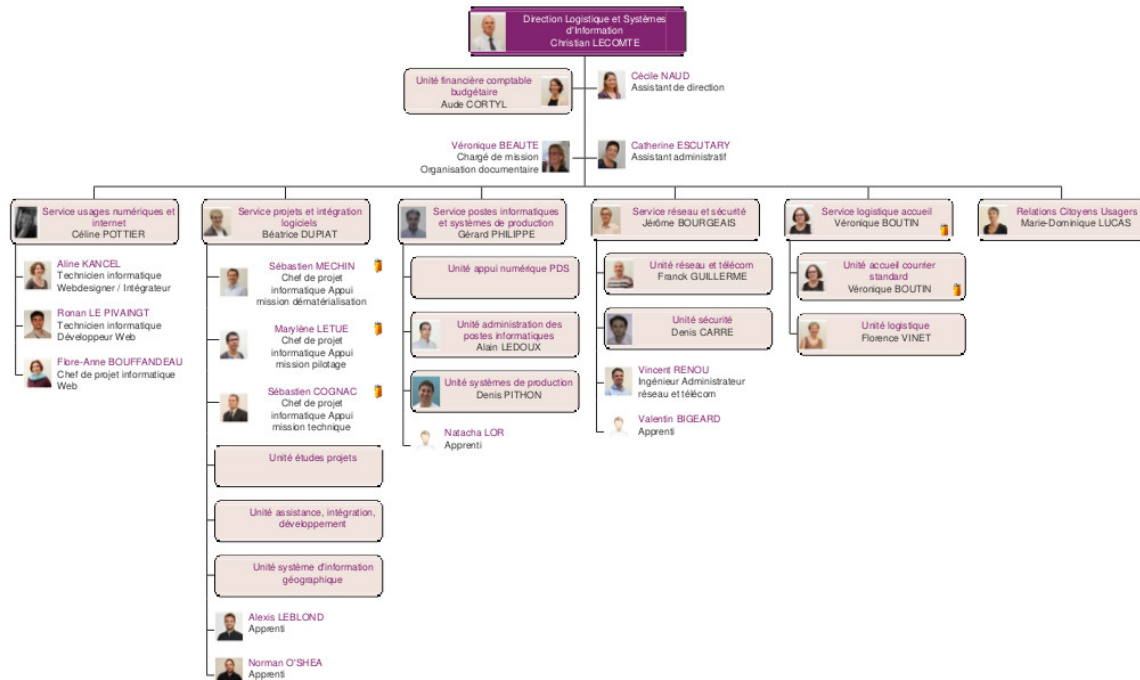


Image 2. Organigramme de la DLSI

# L'unité Système de Production et ses missions

L'unité système de production travaille au sein de la DLSI du Conseil Départemental de Maine-et-Loire sur les problématiques liées au stockage, à la sauvegarde, à la virtualisation et à la supervision des matériels et applications cotés serveurs, ainsi qu'à l'administration des systèmes Linux et Windows.

Quoi ? missions ? chiffres ? → Stockage → Sauvegarde/restauration → Virtualisation → Administration système et bases de données → Gestion des profils/boite mail → SUPERVISION

## Service postes informatiques et systèmes de production

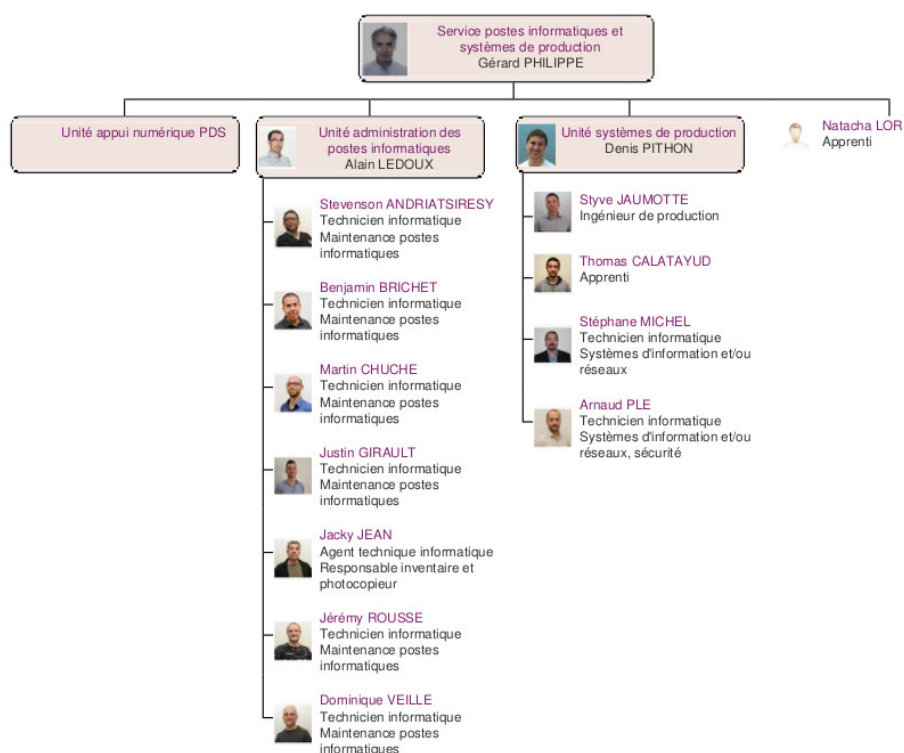


Image 3. Organigramme du Service Poste Informatique et Système de Production

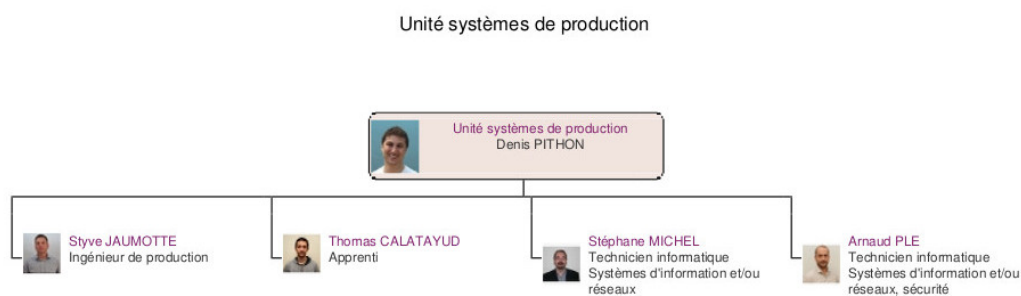


Image 4. Organigramme de l'unité Système de Production

---

## La Supervision

La supervision est une fonction permettant d'indiquer, contrôler, commander l'état d'un système ou d'un réseau. Les outils de supervision remontent des informations techniques et fonctionnelles du système d'information. Le tout dans un but de détection et de traitement le plus automatique possible.

L'informatique est intégrée et est devenue un outil indispensable d'une entreprise, quel que soit son secteur d'activité, le système d'information est placé désormais au centre de l'activité de différentes entités métiers et doit fonctionner plainement et en permanence pour garantir l'efficacité de l'entreprise. A tous les niveaux, les réseaux, les terminaux utilisateurs, les serveurs d'applications et les données constituent autant de maillons sensibles dont la disponibilité et la qualité de service conditionnent le bon fonctionnement de l'entreprise.

Il existe deux enjeux majeurs pour les directions informatiques. Le premier est de garantir la disponibilité et les niveaux de service du système en cas de panne ou de dégradation des performances. Le second est de tenter de prévenir en cas de problème et, le cas échéant, garantir une remontée d'information rapide et une durée d'intervention minimale. Ces enjeux sont donc assurés par la supervision.

Ainsi, la supervision inclut plusieurs activités :

- Surveiller
- Visualiser
- Analyser
- Piloter
- Agir
- Alerter

Elle permet de superviser l'ensemble du Système d'Information d'une entreprise :

- Le réseau et ses équipements
- Les serveurs
- Les périphériques
- Les applications
- Le workflow
- ...



---

## Présentation de l'outil Nagios

Nagios, qui s'appelait précédemment NetSaint, est un outil de supervision libre sous licence GPL. Développé en 1996, Nagios, s'architecture autour d'un moteur écrit en C. Il permet d'auditer en permanence des machines, des services sur ces machines, de recevoir des alertes en cas de problème et de disposer d'un tableau de bord de l'état du système à un moment donnée. C'est un programme modulaire qui se décompose en trois parties :

- Le moteur de l'application qui vient ordonnacer les tâches de supervision.
- L'interface web, qui permet d'avoir une vue d'ensemble du Système d'Information et des éventuelles anomalies.
- Les sondes (ou plugins), une centaine de mini programmes/scripts que l'on peut compléter, voir même créer, en fonction des besoins de chacun pour superviser chaque service ou ressource disponible sur l'ensemble des éléments du réseaux du Système d'Information.

Cet outil offre de nombreuses possibilités :

- Superviser des services réseaux (SMTP, HTTP, ICMP, ...)
- Superviser les ressources des serveurs (charge du processeur, occupation des disques durs, utilisation de la mémoire, ...) sur la majorité des systèmes d'exploitation.
- Superviser les équipements réseau (CPU, ventilateurs, ...)
- Superviser les Bases de données
- Superviser l'environnement (température, luminosité, humidité, climatisation, ...)
- Interface via le protocole SNMP
- Supervision à distance via SSH, tunnel SSL ou agent NRPE.
- Remonter des alertes par mails, sms via un système de notification.
- Gestions d'utilisateurs (accès limité à certains utilisateurs)
- Les plugins sont écrits dans des langages de programmation les plus adaptés à leur tâche : scripts shell (bash, ksh, ...), C++, perl, Python, Ruby, PHP, C#, ... et il est possible de créer les siens.

# Etude et mise en place d'une solution libre/open-source de supervision

Actuellement, la supervision de l'ensemble du système d'information est opérée par Nagios. Cette solution, en place depuis près de 10 ans, contrôle un peu plus de 2700 points de fonctionnement du SI (espaces disques, sites webs, bases de données, consommations CPU, RAM ...).



## *Quelques éléments d'information concernant le système d'information :*

Virtualisation sur oVirt (Linux/KVM)

- ~ 365 VMs (55% Linux, 45% Windows) réparties sur 42 serveurs physiques
- la moitié de ces VMs servent les applications métiers des 2500 agents
- Stockage NAS (NFS et CIFS) répliqué sur deux salles
- 18 To consommés pour les VMs
- 15 To consommés pour la bureautique
- Supervision avec Nagios

## **Il m'est demandé dans le cadre de mon apprentissage de :**

1. Identifier et comparer les solutions libres/open-sources de supervision
2. Préconiser la solution la plus adaptée aux besoins de l'unité
3. Mettre en place la solution de supervision retenue

# Mon activité au sein de l'unité

## Mise à niveau technique

Dans un premier temps, à mon arrivé, il m'a été conseillé de commencer par me mettre à niveau, pour gagner en compétences techniques et monter en puissance sur le système. On m'a donc proposé une série de petits exercices à difficulté progressive. Ils ont pour but de me faire progresser sur l'environnement Linux côté serveur et les outils qui lui sont habituellement associés et me familiariser avec l'administration système pour gagner en autonomie.

Avant tout, il a fallu que j'installe et je configure entièrement mon poste de travail sous Linux.

J'ai ensuite découvert l'outil Ovirt que notre unité utilise pour l'installation et la gestion de machines virtuelles, pour installer et configurer une machine virtuelle Windows.

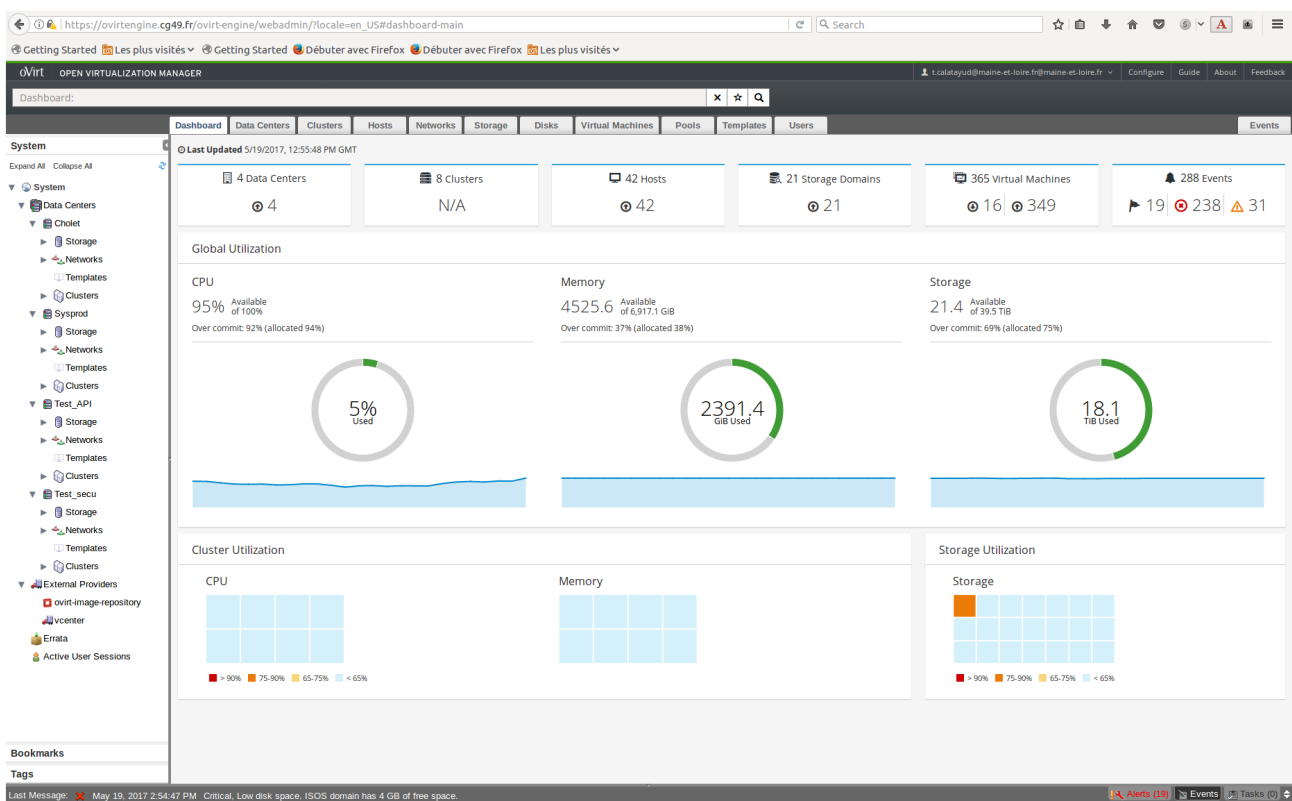


Image 5. Ovirt

## Monter un disque virtuel

On m'a ensuite demandé de construire et monter un disque virtuel de 500 Po sur mon poste. Il a donc fallu que je trouve un moyen de créer un disque réellement utilisable de 500 Po. Il s'est avéré que désormais, le système d'exploitation empêche de manipuler des volumes aussi gros. J'ai donc pu monter un disque d'une taille seulement de 15 To, ce qui reste un disque conséquent.

```
tcalatayud@tcalatayud-CD49:~$ df -lh
...
/dev/loop0      15T  6,3M   15T   1% /media/tcalatayud/e9567653-9578-4332-b449-
37eb63cab7b ①
...
```

- ① J'obtiens donc un disque d'une taille de 15 To sur lequel je peux écrire et lire des fichiers. Cependant, il est bien entendu évident qu'avant de pouvoir le remplir complètement je risque d'avoir quelques problèmes étant limité par la taille du disque dur physique.

## Script d'alertes mail, inotify

Il m'a ensuite été proposé, d'écrire un script permettant de transmettre un fichier par mail lorsqu'il apparaît dans un répertoire donné, puis le supprimer. Il s'agit donc d'effectuer la surveillance d'un répertoire et de rapporter par mail tout ce qu'il s'y est passé.

Dans un premier temps j'ai écrit un premier petit script en shell bash. Ce script transfère par mail un fichier donné en paramètre s'il est dans le répertoire surveillé et il le supprime une fois qu'il a été envoyé.

J'ai ensuite écrit une deuxième version, améliorée, utilisant le mécanisme inotify qui fournit des notifications concernant le système de fichiers. Ce mécanisme permet de mettre en place des actions associées à l'évolution de l'état du système de fichiers. Les principaux événements qui peuvent être suivis sont :

- **IN\_ACCESS** : Le fichier est accédé en lecture
- **IN\_MODIFY** : Le fichier est modifié
- **IN\_CLOSE\_WRITE** : Le fichier est fermé après avoir été ouvert en écriture
- ...

Et enfin, j'ai écrit une dernière version en Python3 qui s'exécute en tant que daemon, c'est à dire que le programme s'exécute en tâche de fond par le système, sans le contrôle de l'utilisateur.

J'y ai inclus la gestion de logs pour qu'on puisse avoir un rapport, si nécessaire, des actions que le script a effectué et pour permettre d'avoir un aperçu du bon fonctionnement et de la bonne exécution du programme.

J'ai notamment utilisé un fichier de configuration .ini qui permet de définir à l'utilisateur et de rassembler des variables dans un même endroit pour pouvoir les utiliser ensuite dans le programme.

```
[config_mail]
fromaddr = t.calatayud@maine-et-loire.fr ①
toaddr = t.calatayud@maine-et-loire.fr ②
server = smtp.cg49.fr ③
port = 25

[config_inotify]
watchFolder = /home/tcalatayud/sendMailPython/dossier ④

[config_daemon]
pidfile = /home/tcalatayud/sendMailPython/daemon.pid ⑤
logfile = /var/log/MyLog/MyScriptDaemon.log ⑥
```

- ① Adresse mail de l'expéditeur
- ② Adresse mail du destinataire
- ③ Serveur smtp

- ④ Chemin du répertoire surveillé
- ⑤ Chemin du fichier où on retrouve l'id du processus
- ⑥ Chemin du fichier de log

Pour l'exécuté, il suffit de lancer le programme avec l'argument "start". On l'arrête avec l'argument "stop". Il est possible d'obtenir les informations concernant le statut du programme avec l'argument "status".

## Découverte, déploiement, installation et configuration de l'outil Nagios

Dans la continuité de cette mise à niveau, j'ai commencé à jeter un oeil sur l'outil nagios, à voir comment il fonctionne, comment l'installer, comment le configurer, comment l'utiliser.



*Image 6. How to install nagios4*

J'ai donc décidé, pour prendre en main cet outil complexe et puissant, d'installer et de configurer ma propre version de Nagios.

Il m'a donc été nécessaire d'abord, de configurer un serveur pour pouvoir le déployer. J'ai donc installé et configuré une nouvelle machine virtuelle sous CentOS grâce à l'outil Ovirt. Puis je me suis lancé dans l'installation du Nagios en suivant la documentation, sur lequel j'ai configuré quelques sondes pour comprendre leurs fonctionnement.

Pour fonctionner, Nagios est basé sur un système de fichiers de configuration. Ces fichiers de configuration sont situés dans le dossier `/usr/local/nagios/etc/` et classés sous forme de contact, d'hôtes, de services et de commandes.

Les contacts sont les différents utilisateurs de l'outils. On peut leur attribuer différents degrés de droit d'accès, définir une adresse mail, les périodes de notifications, ...

Les hôtes sont les différents serveurs, équipements en réseaux que l'on supervise. Il est impératif de lui définir l'adresse IP à laquelle l'hôte est affecté sur le réseau.

Pour chaque hôtes on définit les différents services à superviser en lui précisant la commande à exécuter et en précisant les différents arguments si nécessaire. Ils remontent via les commandes et les checks l'état dans lequel ils sont. Classiquement, on reconnaît trois types d'états pour les services :

- **OK**
- **WARNING**
- **CRITICAL**

Les checks sont les scripts exécutées par les commandes des différents services qui vont permettre de récupérer les données nécessaires pour indiquer et mettre à jours l'état de ces services.

Dans un premier temps il a fallut que je configure un contact pour y déclarer principalement l'adresse mail sur laquelle je compte recevoir les alertes.

*contacts.cfg*

```
define contact{
    contact_name          nagiosadmin          ; Short name of user
    use                   generic-contact       ; Inherit default
    values from generic-contact template (defined above)
    alias                 Nagios Admin         ; Full name of user

    email                 t.calatayud@maine-et-loire.fr ; <<*****
    CHANGE THIS TO YOUR EMAIL ADDRESS *****
    service_notification_period 24x7
    service_notification_options w,u,c,r,f,s
    service_notification_commands notify-service-by-email
    host_notification_period    24x7
    host_notification_options   d,u,r,f,s
    host_notification_commands  notify-host-by-email
}
```

Dans un second temps, j'ai rajouté les différents hôtes que je souhaite superviser. Voici un exemple d'hôtes que j'ai configuré. C'est le serveur hébergeant le site web interne prévu pour les agents du département.

### *hosts/melinfo.cfg*

```
define host {
    use                generic-host                ; Inherit default
values from a template
    host_name          melinfo                      ; The name we're
giving to this host
    alias              Melinfo                      ; A longer name associated
with the host
    address            10.100.49.110                ; IP address of the
host
    hostgroups         linux-servers
    check_interval      5
    retry_interval      1
    check_command       check-host-alive
    max_check_attempts  10
    contact_groups      admins
    register            1
}
```

Ensuite, il faut configurer les services liés à chaque hôte, les données que je souhaite superviser.

### *services/melinfo\_service.cfg*

```
define service {
    use                generic-service
    host_name          melinfo
    service_description HTTP
    check_command       check_http
    notifications_enabled 0
} ①

define service {
    use                generic-service
    host_name          melinfo
    service_description PING
    check_command       check_ping!100.0,20%!400.0,90%
} ②
```

- ① Ce service va utiliser un check, une commande qui va envoyer une requête http dans le but de savoir si oui ou non le serveur web répond.
- ② Ce service va utiliser un check qui va remonter le temps de réponse, le ping. Cette commande est configuré de sorte à ce que si le temps de réponse dépasse 100 ms il se placera dans l'état WARNING. Si le temps de réponse dépasse 400 ms il se placera dans l'état CRITICAL.

Voici quelques exemples de commandes utilisant des checks définis par Nagios.



```

define command{
    command_name    check_http
    command_line    $USER1$/check_http -I $HOSTADDRESS$ $ARG1$
}

define command{
    command_name    check_ping
    command_line    $USER1$/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$ -p 5
}

define command {
    command_name    check_local_disk
    command_line    $USER1$/check_disk -w $ARG1$ -c $ARG2$ -p $ARG3$
}

```

Il est possible avec Nagios de créer des templates ou groupes pour les contacts, hôtes et services. Ces templates permette d'uniformiser les configurations de contacts, hôtes et services qui se ressemblent pour pouvoir les déclarer plus simplement les prochaines fois.

Avec cette méthode de configuration, sous forme de fichiers, il est important pour garder un environnement propre et pour permettre une maintenance plus simple, de bien gérer l'arborescence de tous ces fichiers de configurations. Dans le cas contraire on se retrouve rapidement perdu et on risque de ne plus savoir où à été rangé ce fichier qu'on aimerai bien modifier.

On obtient un Nagios configuré et pret à superviser. On peut désormais se diriger vers l'interface web pour avoir enfin l'aperçu de notre système de supervision. <http://xthomasnagios2/nagios/> (où xthomas2 est la résolution dns de l'adresse ip de mon serveur sur le réseau)

10.100.141.16/nagios/

nagios notification email

Nagios®

General

Home

Documentation

Current Status

Tactical Overview

Map (Legacy)

Hosts

Services

Host Groups

Summary

Grid

Service Groups

Summary

Grid

Problems

Services (Unhandled)

Hosts (Unhandled)

Network Outages

Quick Search:

Reports

Availability

Trends (Legacy)

Alerts

History

Summary

Histogram (Legacy)

Notifications

Event Log

System

Comments

Downtime

Process Info

Performance Info

Scheduling Queue

Configuration

Current Network Status

Last Updated: Fri Jan 13 16:09:41 CET 2017

Updated every 60 seconds

Nagios® Core™ 4.1.1 - www.nagios.org

Logged in as nagiosadmin

View History For all hosts

View Notifications For All Hosts

View Host Status Detail For All Hosts

Host Status Totals

Up Down Unreachable Pending

3 1 0 0

All Problems All Types

1 4

Service Status Totals

Ok Warning Unknown Critical Pending

11 1 0 3 0

All Problems All Types

4 15

Service Status Details For All Hosts

Limit Results: 100

Host	Service	Status	Last Check	Duration	Attempt	Status Information
444.cg@b.fr	HTTP	OK	01-13-2017 16:03:37	23d 3h 52m 12s	1/3	HTTP OK: HTTP/1.1 200 OK - 2791 bytes in 0.082 second response time
	PING	OK	01-13-2017 16:06:56	0d 14h 22m 44s	1/3	PING OK: Packet loss = 0%, RTA = 0.56 ms
localhost	Current Load	OK	01-13-2017 16:06:57	0d 0h 7m 43s	1/4	OK - load average: 0.34, 0.47, 0.27
	Current Users	OK	01-13-2017 16:06:02	23d 5h 25m 10s	1/4	USERS OK - 1 users currently logged in
	HTTP	OK	01-13-2017 16:08:54	2d 13h 20m 25s	1/4	HTTP OK: HTTP/1.1 200 OK - 3302 bytes in 0.005 second response time
	PING	OK	01-13-2017 16:09:01	4d 0h 24m 3s	1/4	PING OK: Packet loss = 0%, RTA = 0.05 ms
	Root Partition	OK	01-13-2017 16:05:25	0d 0h 14m 15s	1/4	DISK OK - free space: / 3651 MB (47% inode=98%):
melibo	SSH	OK	01-13-2017 16:04:46	3d 0h 54m 37s	1/4	SSH OK - OpenSSH_6.6.1 (protocol 2.0)
	Swap Usage	OK	01-13-2017 16:08:45	0d 0h 10m 55s	1/4	SWAP OK - 100% free (1023 MB out of 1023 MB)
	Total Processes	OK	01-13-2017 16:06:32	0d 0h 9m 8s	1/4	PROCS OK: 71 processes with STATE = RSDT
melibo	HTTP	WARNING	01-13-2017 16:04:48	2d 14h 44m 42s	3/3	HTTP WARNING: HTTP/1.0 404 Not Found - 1053 bytes in 0.066 second response time
	PING	OK	01-13-2017 16:03:30	4d 0h 27m 22s	1/3	PING OK: Packet loss = 0%, RTA = 0.58 ms
xthomas2	Check Nagios	CRITICAL	01-13-2017 16:09:13	0d 0h 14m 52s	4/4	(No output on stdout) slider: connect to address 10.100.141.11 port 5666: No route to host
	Check Ping	CRITICAL	01-13-2017 16:07:41	0d 0h 9m 59s	4/4	CRITICAL - Host Unreachable (10.100.141.11)
	Check Users	CRITICAL	01-13-2017 16:04:35	0d 0h 8m 5s	4/4	(No output on stdout) slider: connect to address 10.100.141.11 port 5666: No route to host

Results 1 - 15 of 15 Matching Services

Image 7. Services Nagios

## Synchronisation des nagios de l'unité

SAEIR, installation des baies de stockage et de serveurs. Déplacement dans les salles. Et installation du nagios.

Rapport d'apprentissage Master 1 - Informatique - Supervision open-source d'un système d'information

16

---

# Identification des solutions libres/open-sources de supervision

- Problématique :
- Solution vieillissante
- Problème de maintenance
- mise à jour ardue (BSD, versions, ...)

Vu le manque de réactivité du développeur principal de Nagios et sa volonté de ne plus diffuser tous les modules sous licence libre, de nombreux développeurs actifs sur le projet ont fait diverger Nagios. Ainsi, de nombreux outils similaire à Nagios ont été créés. S'ouvre donc l'éventualité de voir et découvrir ces nouvelles solutions.

## Instantané des solutions libres/open-sources.

Premier critère : Libre/Open-source

- Inventaire exhaustif
- Donner les différents types de solutions
- Les protocoles utilisés
- Les types de configuration
- Remontée des alertes
- ...

Pour trouver une nouvelle solution de supervision adaptée au besoin de l'unité, il est nécessaire de connaître les enjeux de la supervision, de connaître les solutions existantes sur le marché et ce qu'elles proposent.

Inventaire le plus exhaustif possible, recherche et exploration la plus complète possible et la plus "naïve" possible, avec un regard neuf sur la supervision. En expliquant pourquoi je fais ça. Pourquoi je n'utilise pas de cahier des charges.

→ apparitions des premiers critères évidents ( snmp, plugins, alertes, ... ), éliminations des solutions qui sont vraiment pas adaptés, premier filtrage.

---

## Etablir les critères de sélection

- Identifier le domaine à superviser (sur le nagios)
- Lister les sondes
- Croiser les éléments ressorti lors de l'inventaire
- Dresser la liste des critères

Comment je les ai établi (recherche plus poussée sur les solutions les plus intéressantes, études des sondes sur le nagios actuel), pourquoi ils sont nécessaires, à quoi ils servent, catégorisation, listes avec explication et appréciations, ...

---

# Présentation des solutions sélectionnées

Présentation des 2 solutions : carte d'identité des solutions

## Solution 1

## Solution 2

# Bilan

Synthèse et ouverture Dire ce que j'ai fais durant cette première année, le déroulement. (Missions annexes) Mon ressenti ce que j'ai apporté, ce qu'on m'a apporté

---

# Annexes

---

# Table des illustrations

## Table des illustrations

---

# Sources