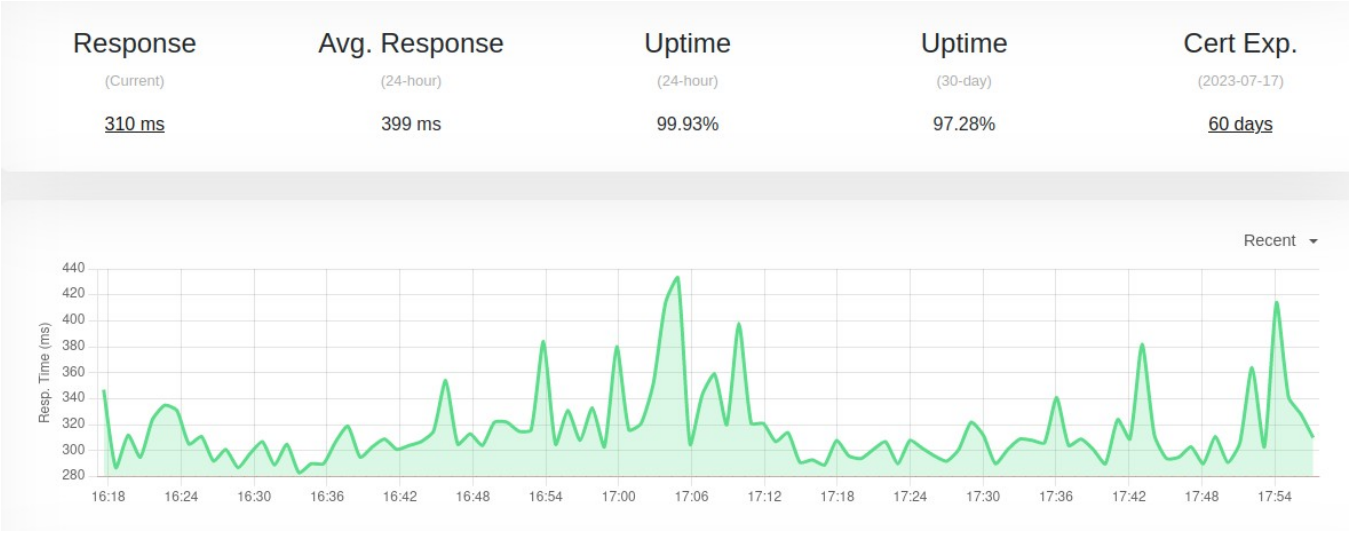
Supervision et métrologie



Introduction

Introduction

- **Objectifs du cours :**
  - Comprendre les notions de ***« supervision »*** et de ***« métrologie »***
  - Comprendre les enjeux de la supervision et de la métrologie dans le ***contexte*** d'une entreprise, d'un point de vue technique comme d'un point de vue économique
  - Être capable d'identifier les ***assets*** à superviser et de sélectionner les méthodes et outils adaptés
  - Partager des ***retours d'expériences*** et débattre sur des cas d'usage ou des approches

Introduction

- **Plan de cours :**
  - La supervision « traditionnelle »
    - Pourquoi superviser ? 🤔
    - Protocoles utilisés
    - Cas d’usage
  - La métrologie
  - La supervision « étendue »
    - Logs, sécurité, réseau…
  - Gouvernance et « hypervision »

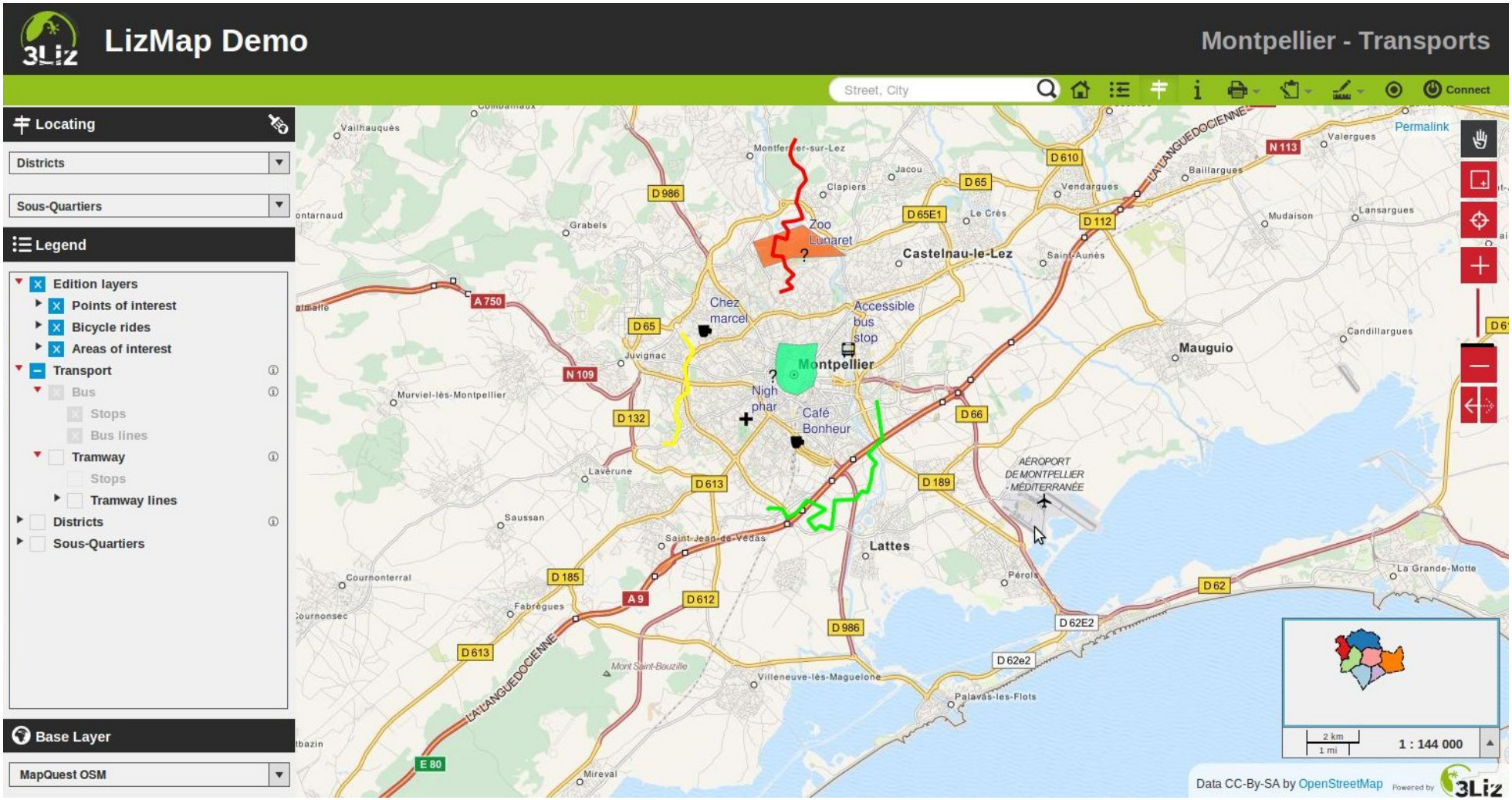
Introduction

- **Évaluations :**
  - 1 QCM (15 min) à la fin de chaque cours
    - 3 notes sur 15
  - 1 note de TP
    - 1 note sur 15 répartie en plusieurs petits exercices

Introduction

- **Qui suis-je ?**
  - Sysadmin, netadmin, gestion de projet, formation
  - Aujourd'hui chez 3Liz, éditeur Open Source de SIG et hébergeur SaaS
  - Plusieurs expériences professionnelles :
    - FAI, équipementier, sécurité/IAM, SSII, hébergeur/infogéreur, Cloud provider…
  - Plusieurs centres d'intérêts :
    - Stockage, sauvegarde, réseau, logiciel libre…
    - Les trains, Snowrunner, WoW, les 4X, le tir à l'arc et la cuisine (végétarienne/végane)

Introduction



La supervision

Supervision : études de cas

Etudes de cas : énoncé

- **Contexte :**
  - 4 profils d’entreprise = 4 groupes
- **Objectifs :**
  - Lister les assets à superviser (listes non exhaustives !)

- **Vous avez 15 min** 👍

Vos 4 entreprises

- **1 : TPE/PME industrielle** 
  - 1 usine de production avec machines-outils et hangars, 2 agences de bureaux, des connexions directes avec les fournisseurs/clients/transporteurs pour l'optimisation des flux de production
- **2 : Centre Hospitalier** 
  - 3 sites, 80 serveurs, 500 PC (administratif et salles d'examens), 2000 équipements médicaux
- **3 : Fournisseur d'accès à Internet** 
  - 2 datacentres, 9 POP, 4 transitaires, 4000 fibres/xDSL
- **4 : Cloud Provider** 
  - 20 datacentres, 200000 serveurs, 1,5 millions d'instances VPS, 1 million de clients

Etude de cas : entreprise 1

- **TPE/PME industrielle**
  - **Les PC de contrôle + les machines-outils (télémaintenances tierces ?)**
  - **L'état des liaisons avec les partenaires (tunnels de commandes, VPN, API…)**
  - Les éléments de sécurité (caméras, alarmes, détecteurs…)
  - Les switchs et bornes Wifi de l'usine, des hangars et des bureaux
  - Les serveurs applicatifs et de gestion du parc (si existants)
  - ***Les PC des employés ?***

  - **Un arrêt des tunnels de commandes = un arrêt de production = perte financière**

Etude de cas : entreprise 2

- **Centre hospitalier**
  - Les serveurs et VM de gestion du parc (applications métiers/médicales)
  - Les switchs et bornes Wifi (roaming appareils médicaux)
  - **Les téléphones et systèmes d'alertes des personnels médicaux**
  - **Les PC des salles d'examens et d'opérations**
  - **Les équipements médicaux (télémaintenance avec une entreprise spécialisée)**
  - ***Le site internet de l'hôpital ?***

  - **Un équipement médical indisponible ou un PC de salle d'opération trop long à ouvrir une session = des vies en jeu**

Etude de cas : entreprise 3

- **Fournisseur d'accès à Internet**
  - Les serveurs et VM de gestion du parc
  - **Les équipements réseaux :**
    - Cœur de réseau
    - Équipements de collecte sur les POP
    - Équipements de livraison chez les clients
  - **Les liens avec les transitaires (sessions BGP)**
  - **La latence vers différents points vitaux d'Internet (serveurs racine DNS, NTP…)**
  - ***L'état des zpools des serveurs de sauvegarde ?***

Etude de cas : entreprise 4

- **Cloud Provider**
  - Le parc de serveurs physiques (état du matériel, état des stocks…)
  - Le taux d'occupation et les capacités électriques et de refroidissement des salles
  - Les plateformes de virtualisation (état, capacité…)
  - Les connexions vers l'extérieur (transits, peering, liens inter-datacentres…)
  - **L'état des sites web hébergés (codes de retours, temps de réponse…)**
  - **Les délais de livraisons des instances VPS aux clients**
  - ***L'état des barrettes mémoires des serveurs ?***

Bilan de l’étude de cas

- **Ne pas se focaliser uniquement sur l’IT !**

- Garder en tête :
  - Le but de l’entreprise (***qu’est-ce que nous livrons à nos clients ?***)
  - La qualité de service (***est-ce que le service est rendu dans de bonnes conditions ?***)
  - Les moyens à mettre en œuvre pour superviser les assets (***coûts financiers, techniques et humains***)

La supervision : pourquoi ?

Définition

*« La supervision est une technique **industrielle** de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables.*

Dans l'informatique, la supervision est la surveillance du bon fonctionnement d'un système ou d'une activité. *»*

Source : Wikipédia

Pourquoi superviser ?

(pour le plaisir de se faire réveiller la nuit en astreinte !) 😭

- D'un point de vue **technique** :
  - **Valider l'état de fonctionnement** d'une machine ou d'un service
  - Anticiper le **dépassement des capacités** d'un environnement
  - Anticiper les **pannes** et les remplacements de matériel, ou les mises à jour applicatives

- D'un point de vue **fonctionnel** :
  - **Valider le service rendu** aux clients (SLA…)
  - Mieux gérer et anticiper les **budgets** (achats, formations, JH…) 

La supervision : comment ?

Comment superviser ?

- Plusieurs types de supervision :
  - **Système** (CPU, RAM…)
  - **Réseau** (débits, disponibilité…)
  - **Applicative** (validation fonctionnelle)

- Utilisation de différents protocoles et de différentes méthodologies selon les besoins
- Les protocoles « classiques » sont pensés à la fois pour **superviser et administrer**

- **Une bonne supervision requiert un bon *inventaire* !**

Une histoire d’assets…

Qu’est-ce qu’un asset ?

- Un asset est un **composant du système d’information**
- Un asset possède une **valeur** pour l’entreprise, des **attributs** et un **cycle de vie**

- Quelques exemples d’assets :
  - Un équipement réseau
  - Un disque dur
  - Un nom de domaine
  - Une licence Minecraft
  - ***Un employé ?***
  - ***Un conteneur Docker ?***

La CMDB, le coeur du SI

- **CMDB : Configuration Management Database**
- **Base de données** fondamentale d'une architecture ITIL unifiant tous les assets d'un système d'informations :
  - **Inventaire** (équipements, licences…)
  - Disponibilité et **utilisation des assets**
  - Attributs et configurations (peut entrer en conflit avec les outils de Configuration Management !)
  - **Liens entre les assets** (câblage, dépendances…)
  - **Historique et suivi** (modifications, pannes…)

- **Peut être composée de plusieurs outils complémentaires (/!\)**
- **Pas de CMDB = supervision incomplète !**

☝️

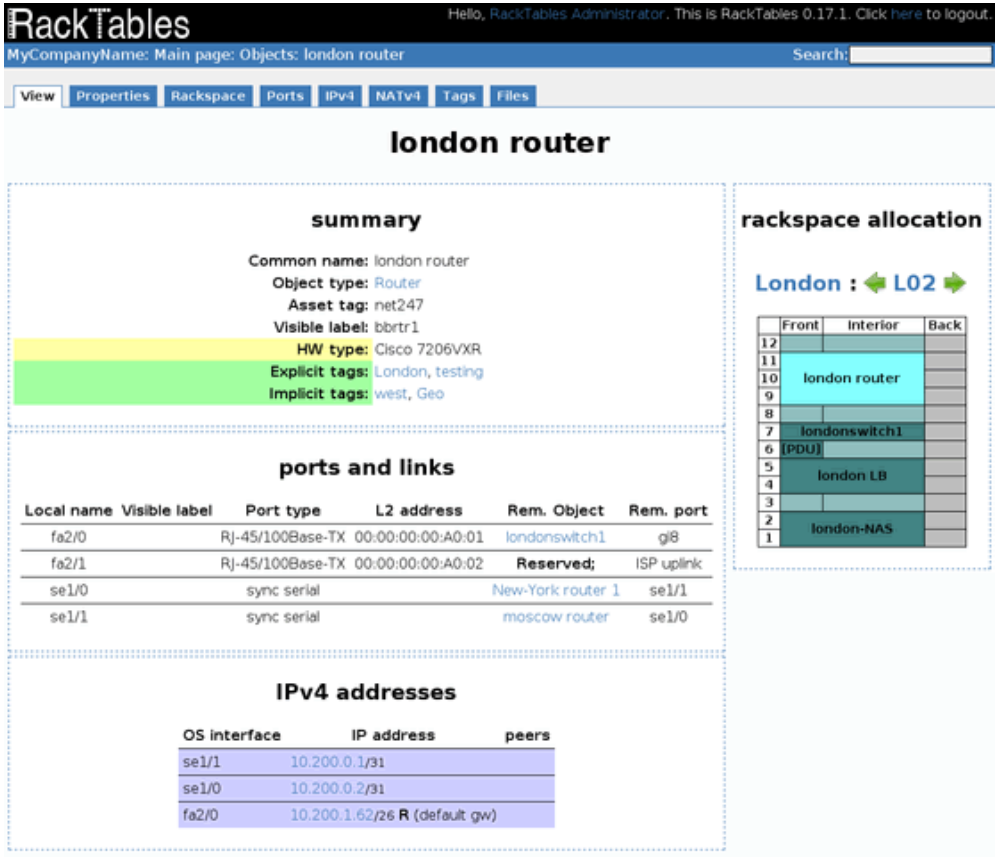
CMDB, DCIM, IPAM…

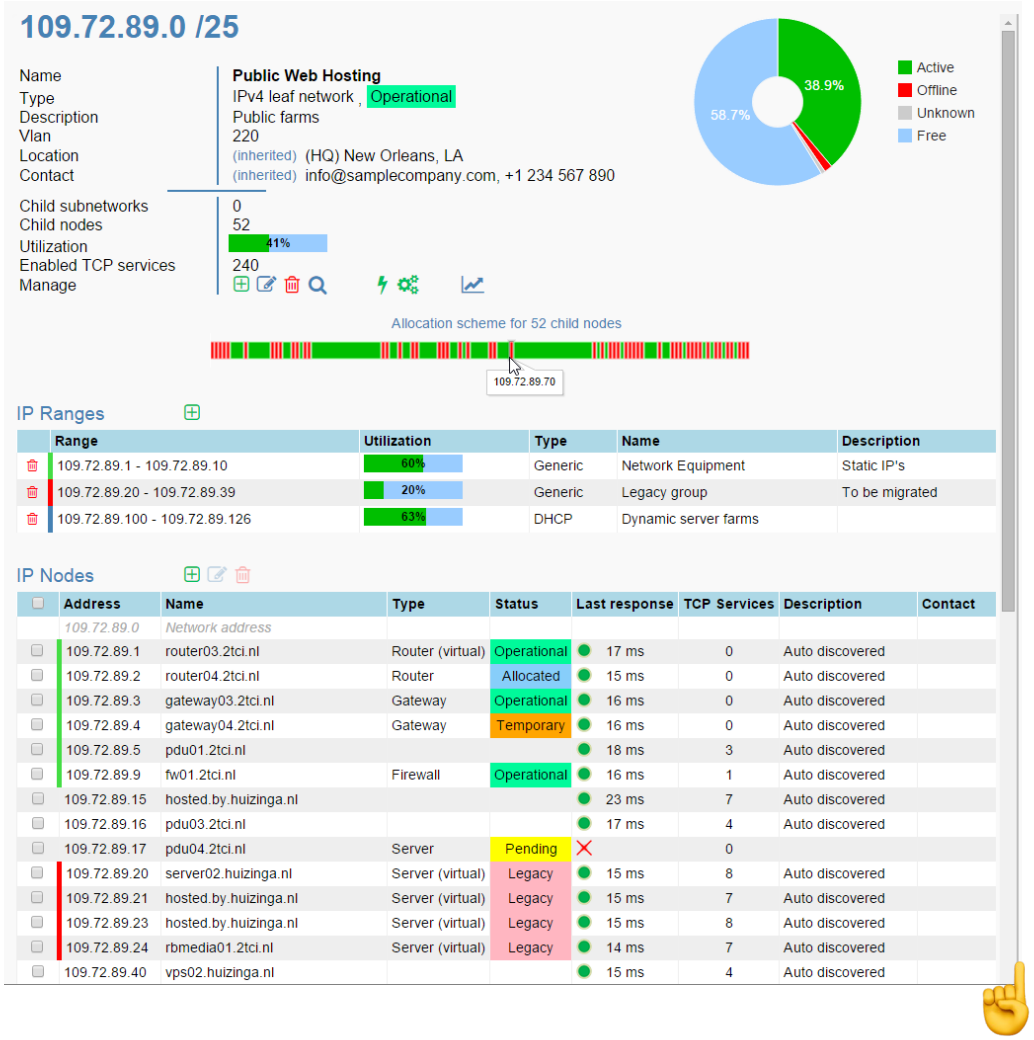
- **IPAM : IP Address Management**
- **ITSM : IT Asset Management**
- **DCIM : Datacenter Infrastructure Management**

- Chaque outil est adapté à un **besoin métier**
- **Un outil qui couvre tous les aspects d'un SI n'existe pas !**

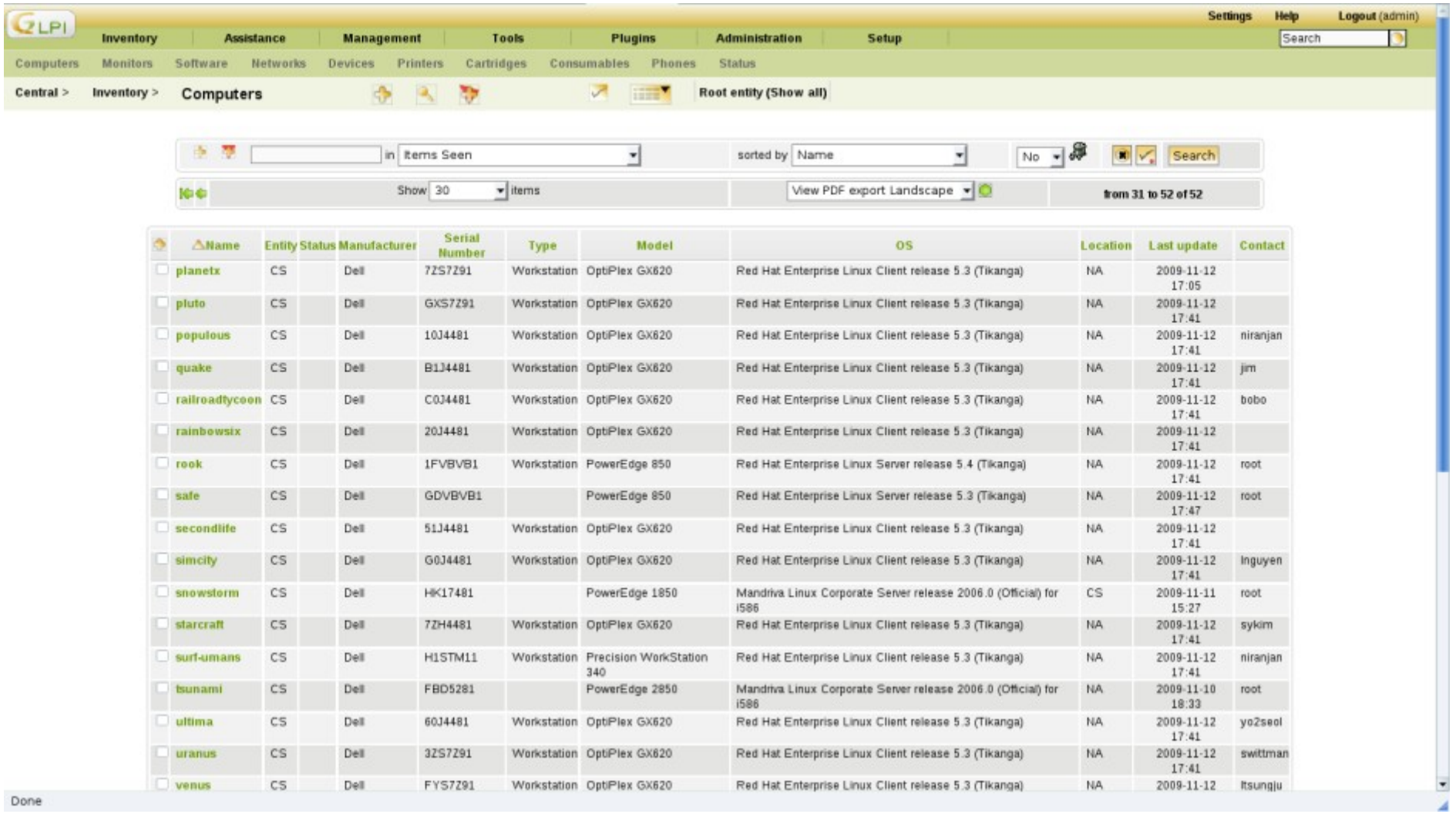
- Beaucoup d'outils disponibles :
  - GLPI
  - RackTables
  - iTop
  - NetBox
  - …

Quelques outils de gestion d'assets





Quelques outils de gestion d'assets



Name	Entity	Status	Manufacturer	Serial Number	Type	Model	OS	Location	Last update	Contact
planetx	CS		Dell	7ZS7Z91	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:05	
pluto	CS		Dell	GXS7Z91	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	
populous	CS		Dell	10J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	niranjan
quake	CS		Dell	B1J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	jim
railroadtycoon	CS		Dell	C0J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	bobo
rainbowsix	CS		Dell	20J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	
rook	CS		Dell	1FVBVB1	Workstation	PowerEdge 850	Red Hat Enterprise Linux Server release 5.4 (Tikanga)	NA	2009-11-12 17:41	root
safe	CS		Dell	GDVBVB1		PowerEdge 850	Red Hat Enterprise Linux Server release 5.3 (Tikanga)	NA	2009-11-12 17:47	root
secondlife	CS		Dell	51J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	
simcity	CS		Dell	G0J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	lnguyen
snowstorm	CS		Dell	HK17481		PowerEdge 1850	Mandriva Linux Corporate Server release 2006.0 (Official) for i586	CS	2009-11-11 15:27	root
starcraft	CS		Dell	7ZH4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	sykim
surf-umans	CS		Dell	H1STM11	Workstation	Precision WorkStation 340	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	niranjan
tsunami	CS		Dell	FBD5281		PowerEdge 2850	Mandriva Linux Corporate Server release 2006.0 (Official) for i586	NA	2009-11-10 18:33	root
ultima	CS		Dell	60J4481	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	yo2seol
uranus	CS		Dell	3ZS7Z91	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12 17:41	swittman
venus	CS		Dell	FYS7Z91	Workstation	OptiPlex GX620	Red Hat Enterprise Linux Client release 5.3 (Tikanga)	NA	2009-11-12	ltsungju

… et de protocoles 😓

Supervision et protocoles : SNMP

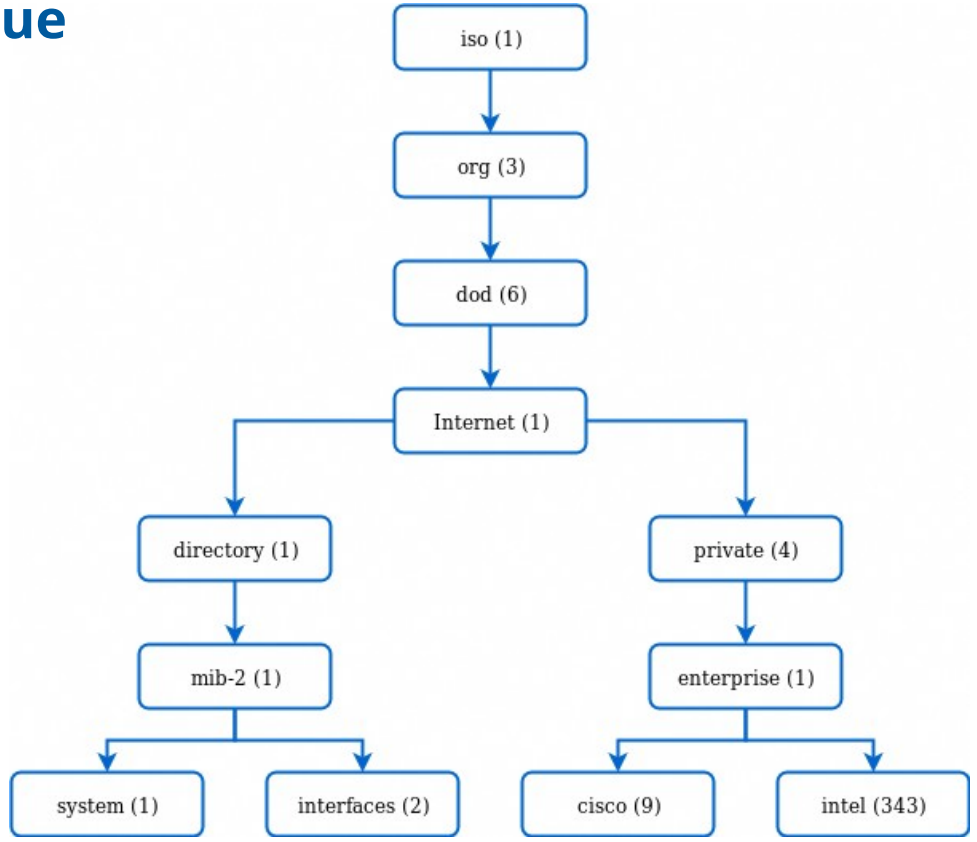
« Simple Network Management Protocol (abrégé SNMP), en français « protocole simple de gestion de réseau », est un protocole de communication qui permet aux administrateurs réseau de ***gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux*** *et matériels à distance. »*

Source : Wikipédia

- Créé en 1988
- RFC 1067 et 1157
- **Ports UDP 161 et 162** 👈

OID (Object Identifier)

- Structure d'identification **hiérarchique**
- Standard de l'ITU-T et de l'ISO
- **Un « OID » = un nœud de l'arbre**
- Chaque nœud est contrôlé par un organisme ou une entreprise



- Exemples :
  - 1 : ISO
  - 1.3.6 : Department of Defense
  - 1.3.6.1.4.1.343 : Intel Corporation

ASN.1 et SMI

- ASN.1 : Abstract Syntax Notation One
  - Standard pour décrire la structure des données

- **SMI : Structure of Management Information**
  - Adaptation d'ASN.1 pour définir les objets d'une MIB

- 3 exemples d'attributs standardisés :
  - ***Objects Identifiers*** (forme numérique ou littérale)
    - *1.3.6.1.2.1* ou *iso.org.dod.internet.mgmt.mib-2*
  - ***Objects Types*** : integer, string, IP address…
  - ***Object Encoding Method*** (comment transmettre cette donnée)

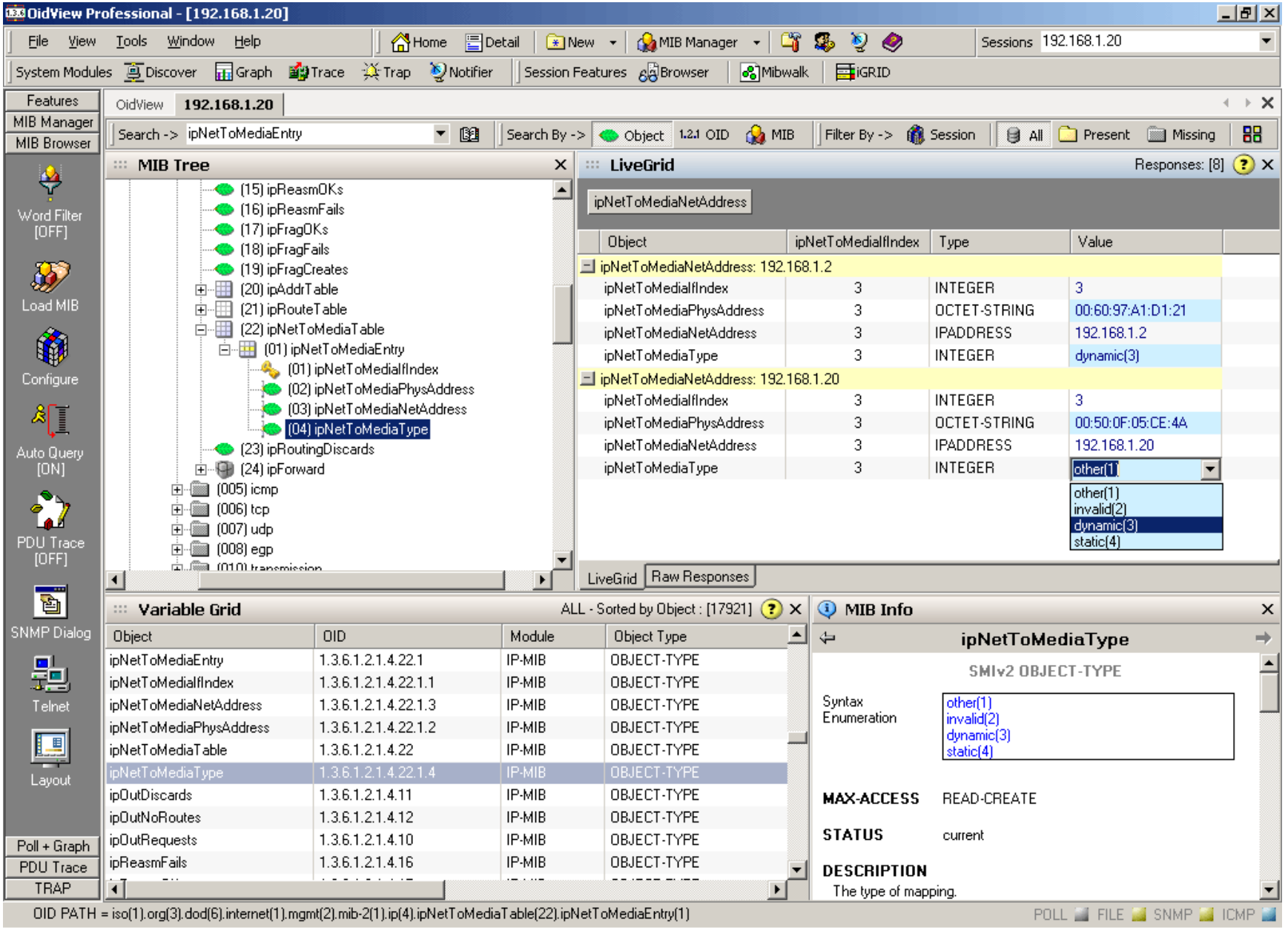
MIB (Management Information Base)

- **Structure de données hiérarchique**
- Chaque entrée est associée à un OID
- Souvent définie sous forme de **« modules »**
  - Plus de 300 MIBs standardisées (SNMPv2-MIB, IP-MIB…)
  - Et les autres sont créées et maintenues par les fabricants

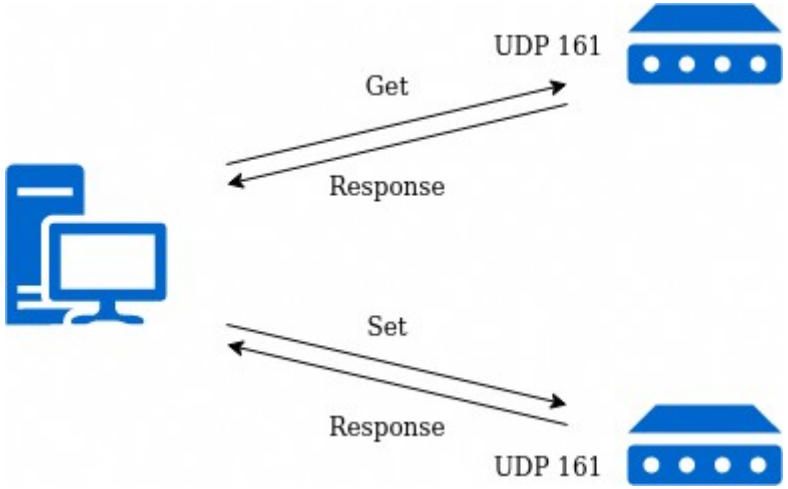
- Accessible via un agent côté client

- Pour trouver l'OID qui nous intéresse, il est parfois nécessaire d'utiliser un « MIB Browser »

Une MIB en image



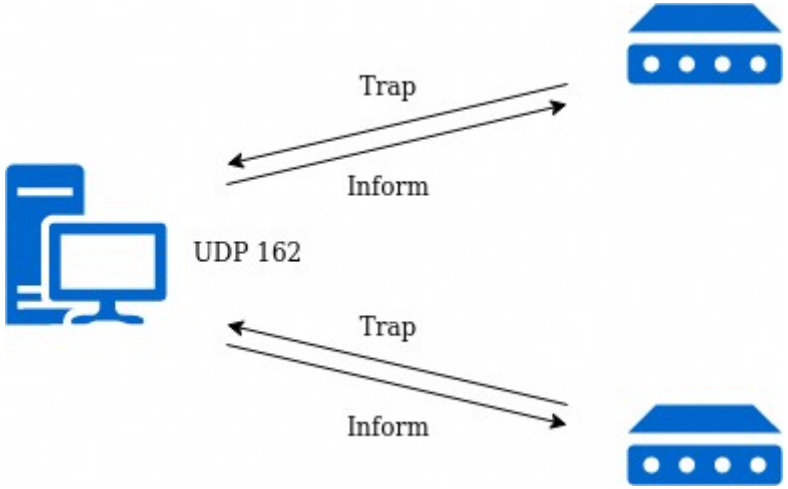
SNMP : architecture client-serveur



- **Schéma d'une PDU :**

Version	Community	PDU Type	Request ID	Error Status	Error Index	Variable Binding List

SNMP : traps



- **Schéma d'une PDU de type « trap » :**

Version	Community	PDU Type	Enterprise Object Identifier	Network Address	Trap Type	Specific Trap Type	Timestamp	Variable Binding List

SNMP v1

- **Compteurs 32 bits**
- **Echanges en clair (PDUs et communautés)**
- 5 PDUs :
  - *GetRequest* : demande une variable au client
  - *SetRequest* : définit une variable (obtient automatiquement une *Response* en retour)
  - *Response* : envoie une variable au serveur
  - *GetNextRequest* : demande la variable suivante au client
  - *Trap* : envoie spontanément une variable et l'OID correspondant au server

SNMP v2(c)

- **Compteurs 64 bits**
- **Echanges en clair (PDUs et communautés)**
- 2 nouvelles PDUs :
  - *GetBulkRequest* : demande une série de variables au client
  - *InformRequest* : informe le client de la réception de son *Trap*

- ***Attention : format incompatible avec SNMPv1, nécessite une dual-stack ou un proxy***

SNMP v3

- **Focus sur la sécurité (authentification, confidentialité et intégrité)**, 3 modes d'utilisation :
  - *NoAuthNoPriv*
  - *AuthNoPriv*
  - *AuthPriv*
- MD5 et SHA pour l'authentification, CBC_DES and CFB_AES_128 pour le chiffrement
- 1 nouvelle PDU :
  - *Report* : utilisée comme message « d'erreur »
- Reconnu comme standard par l'IETF depuis 2004

SNMPwalk en image

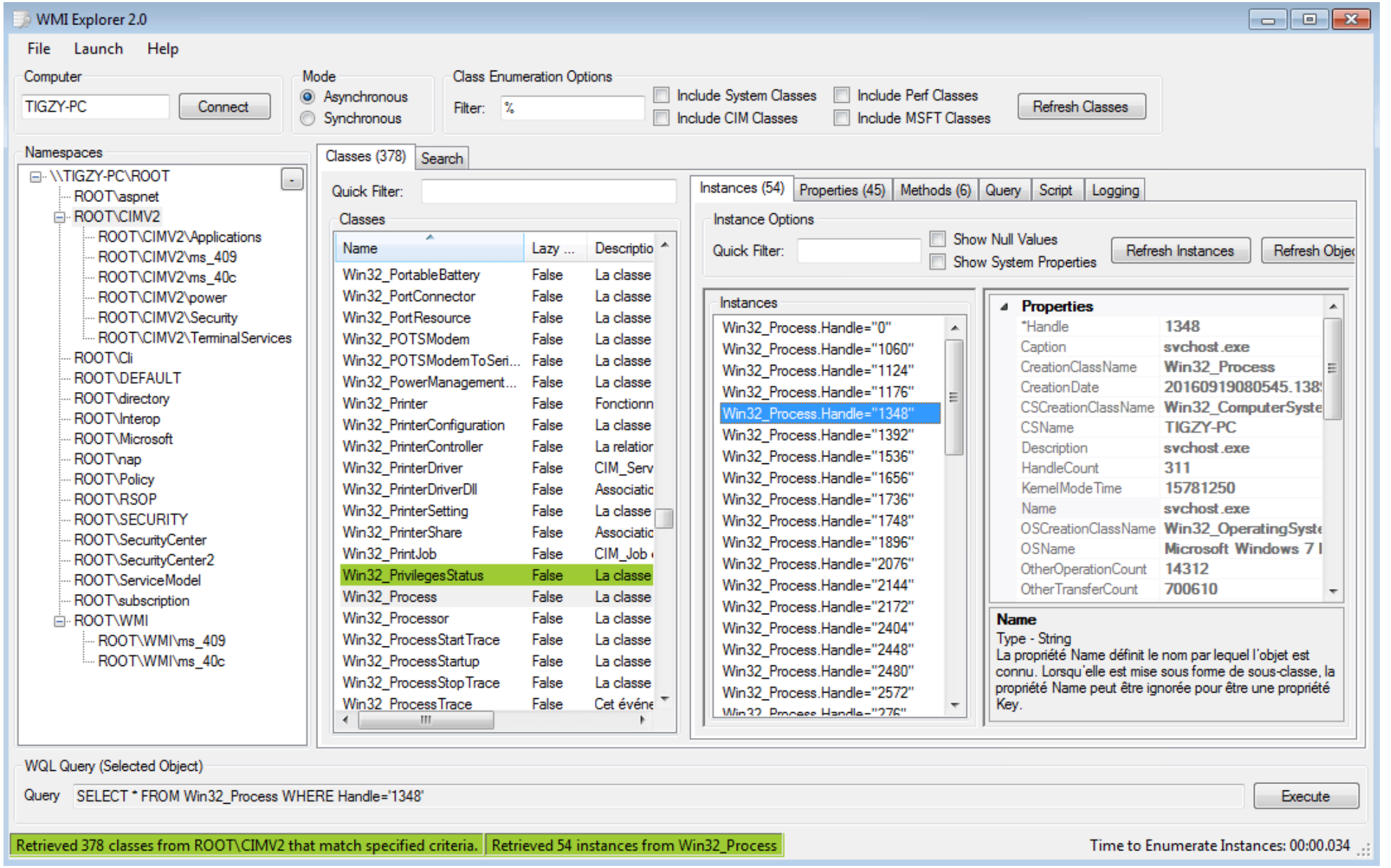
```
librenms@librenms:~$ snmpwalk -v 2c -c toto 127.0.0.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux librenms 5.3.13-1-pve #1 SMP PVE 5.3.13-1 (Thu, 05 Dec 2019 07:18:14 +0100) x86_64"
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (602) 0:00:06.02
iso.3.6.1.2.1.1.4.0 = STRING: "admin@toto.fr"
iso.3.6.1.2.1.1.5.0 = STRING: "librenms"
iso.3.6.1.2.1.1.6.0 = STRING: "OVH"
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.11.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.15.2.1.1
iso.3.6.1.2.1.1.9.1.2.3 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.4 = OID: iso.3.6.1.6.3.1
iso.3.6.1.2.1.1.9.1.2.5 = OID: iso.3.6.1.6.3.16.2.2.1
iso.3.6.1.2.1.1.9.1.2.6 = OID: iso.3.6.1.2.1.49
iso.3.6.1.2.1.1.9.1.2.7 = OID: iso.3.6.1.2.1.4
iso.3.6.1.2.1.1.9.1.2.8 = OID: iso.3.6.1.2.1.50
iso.3.6.1.2.1.1.9.1.2.9 = OID: iso.3.6.1.6.3.13.3.1.3
iso.3.6.1.2.1.1.9.1.2.10 = OID: iso.3.6.1.2.1.92
iso.3.6.1.2.1.1.9.1.3.1 = STRING: "The MIB for Message Processing and Dispatching."
iso.3.6.1.2.1.1.9.1.3.2 = STRING: "The management information definitions for the SNMP User-based Security Model."
iso.3.6.1.2.1.1.9.1.3.3 = STRING: "The SNMP Management Architecture MIB."
iso.3.6.1.2.1.1.9.1.3.4 = STRING: "The MIB module for SNMPv2 entities"
iso.3.6.1.2.1.1.9.1.3.5 = STRING: "View-based Access Control Model for SNMP."
iso.3.6.1.2.1.1.9.1.3.6 = STRING: "The MIB module for managing TCP implementations"
iso.3.6.1.2.1.1.9.1.3.7 = STRING: "The MIB module for managing IP and ICMP implementations"
iso.3.6.1.2.1.1.9.1.3.8 = STRING: "The MIB module for managing UDP implementations"
iso.3.6.1.2.1.1.9.1.3.9 = STRING: "The MIB modules for managing SNMP Notification, plus filtering."
iso.3.6.1.2.1.1.9.1.3.10 = STRING: "The MIB module for logging SNMP Notifications."
iso.3.6.1.2.1.1.9.1.4.1 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.2 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.3 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.4 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.5 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.6 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.7 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.8 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.9 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.4.10 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.2.1.0 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.1.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.1.32 = INTEGER: 32
iso.3.6.1.2.1.2.2.1.2.1 = STRING: "lo"
iso.3.6.1.2.1.2.2.1.2.32 = STRING: "eth0"
iso.3.6.1.2.1.2.2.1.3.1 = INTEGER: 24
iso.3.6.1.2.1.2.2.1.3.32 = INTEGER: 6
iso.3.6.1.2.1.2.2.1.4.1 = INTEGER: 65536
iso.3.6.1.2.1.2.2.1.4.32 = INTEGER: 1500
iso.3.6.1.2.1.2.2.1.5.1 = Gauge32: 10000000
iso.3.6.1.2.1.2.2.1.5.32 = Gauge32: 4294967295
iso.3.6.1.2.1.2.2.1.6.1 = ""
iso.3.6.1.2.1.2.2.1.6.32 = Hex-STRING: FE C3 74 F7 DC 5C
```

Supervision et protocoles : WMI

- **WMI : Windows Management Instrumentation**
  - Apparu avec Windows NT 4.0 (1996)
  - Permet de récupérer des informations sur des composants du système, des applications…
  - … et permet de les configurer (CLI, scripts VBScript ou PowerShell, consoles GUI…)

- Permet aux développeurs d'exposer un **« modèle » (objets et relations)** à travers un **« provider »** WMI

WMI en image


WMI Explorer 2.0
File
Launch
Help
Computer
TIGZY-PC
Connect
Mode
Asynchronous
Synchronous
Class Enumeration Options
Filter:
%
Include System Classes
Include CIM Classes
Include Perf Classes
Include MSFT Classes
Refresh Classes
Namespaces
\\TIGZY-PC\ROOT
ROOT\aspnet
ROOT\CIMV2
ROOT\CIMV2\Applications
ROOT\CIMV2\ms_409
ROOT\CIMV2\ms_40c
ROOT\CIMV2\power
ROOT\CIMV2\Security
ROOT\CIMV2\TerminalServices
ROOT\Cli
ROOT\DEFAULT
ROOT\directory
ROOT\Interop
ROOT\Microsoft
ROOT\nap
ROOT\Policy
ROOT\RSOP
ROOT\SECURITY
ROOT\SecurityCenter
ROOT\SecurityCenter2
ROOT\ServiceModel
ROOT\subscription
ROOT\WMI
ROOT\WMI\ms_409
ROOT\WMI\ms_40c
Classes (378)
Search
Quick Filter:
Classes
Name
Lazy ...
Descriptio
Win32_PortableBattery False La classe
Win32_PortConnector False La classe
Win32_PortResource False La classe
Win32_POTSModem False La classe
Win32_POTSModemToSeri... False La classe
Win32_PowerManagement... False La classe
Win32_Printer False Fonctionn
Win32_PrinterConfiguration False La classe
Win32_PrinterController False La relatior
Win32_PrinterDriver False CIM_Serv
Win32_PrinterDriverDll False Associatic
Win32_PrinterSetting False La classe
Win32_PrinterShare False Associatic
Win32_PrintJob False CIM_Job
Win32_PrivilegesStatus False La classe
Win32_Process False La classe
Win32_Processor False La classe
Win32_ProcessStartTrace False La classe
Win32_ProcessStartup False La classe
Win32_ProcessStopTrace False La classe
Win32_ProcessTrace False Cet événe
Instances (54)
Properties (45)
Methods (6)
Query
Script
Logging
Instance Options
Quick Filter:
Show Null Values
Show System Properties
Refresh Instances
Refresh Objec
Instances
Win32_Process.Handle="0"
Win32_Process.Handle="1060"
Win32_Process.Handle="1124"
Win32_Process.Handle="1176"
Win32_Process.Handle="1348"
Win32_Process.Handle="1392"
Win32_Process.Handle="1536"
Win32_Process.Handle="1656"
Win32_Process.Handle="1736"
Win32_Process.Handle="1748"
Win32_Process.Handle="1896"
Win32_Process.Handle="2076"
Win32_Process.Handle="2144"
Win32_Process.Handle="2172"
Win32_Process.Handle="2404"
Win32_Process.Handle="2448"
Win32_Process.Handle="2480"
Win32_Process.Handle="2572"
Properties
*Handle 1348
Caption svchost.exe
CreationClassName Win32_Process
CreationDate 20160919080545.138
CSCreationClassName Win32_ComputerSyste
CSName TIGZY-PC
Description svchost.exe
HandleCount 311
KernelModeTime 15781250
Name svchost.exe
OSCreationClassName Win32_OperatingSyste
OSName Microsoft Windows 7 I
OtherOperationCount 14312
OtherTransferCount 700610
Name
Type - String
La propriété Name définit le nom par lequel l'objet est connu. Lorsqu'elle est mise sous forme de sous-classe, la propriété Name peut être ignorée pour être une propriété Key.
WQL Query (Selected Object)
Query
SELECT * FROM Win32_Process WHERE Handle='1348'
Execute
Retrieved 378 classes from ROOT\CIMV2 that match specified criteria.
Retrieved 54 instances from Win32_Process
Time to Enumerate Instances: 00:00.034

Supervision et protocoles : IPMI

- **IPMI : Intelligent Platform Management Interface**
  - Introduit par Intel en 1998, utilisé par plus de 200 fabricants aujourd'hui
  - Fournit des interfaces pour **superviser et configurer « out-of-band » des éléments indépendants du système d'exploitation** (BIOS, cartes-filles, firmwares…)
  - Un module IPMI est organisé autour d'un Baseboard Management Controller (BMC), ainsi que de capteurs (températures, vitesses de rotation des ventilateurs…) et de microcontrôleurs

- Exemples d'interfaces exploitant IPMI :
  - **iDRAC** (Integrated Dell Remote Access Controller) chez Dell
  - **iLO** (Integrated Lights-Out) chez HPE

Un exemple de console utilisant IPMI

Integrated Dell Remote Access Controller 7 | Enterprise | Support | About | Logout

System
PowerEdge R920
root , Admin

- Overview
  - Server
    - Logs
    - Power / Thermal
    - Virtual Console
    - Alerts
    - Setup
    - Troubleshooting
    - Licenses
    - Intrusion
  - iDRAC Settings
  - Hardware
  - Storage

Power Monitoring | Power Configuration | Voltages | Temperatures

Temperatures

Fresh Air

Not a Fresh Air Compliant Configuration

Temperature Probes

			Warning Threshold		Critical Threshold	
Status	Probe Name	Reading	Min	Max	Min	Max
✓	CPU1 Temp	37 °C (98.6 °F)	8 °C (46.4 °F)	78 °C (172.4 °F)	3 °C (37.4 °F)	83 °C (181.4 °F)
✓	CPU2 Temp	38 °C (100.4 °F)	8 °C (46.4 °F)	78 °C (172.4 °F)	3 °C (37.4 °F)	83 °C (181.4 °F)
✓	CPU3 Temp	41 °C (105.8 °F)	8 °C (46.4 °F)	78 °C (172.4 °F)	3 °C (37.4 °F)	83 °C (181.4 °F)
✓	CPU4 Temp	37 °C (98.6 °F)	8 °C (46.4 °F)	78 °C (172.4 °F)	3 °C (37.4 °F)	83 °C (181.4 °F)
✓	System Board Exhaust Temp	37 °C (98.6 °F)	8 °C (46.4 °F)	70 °C (158.0 °F)	3 °C (37.4 °F)	75 °C (167.0 °F)
✓	System Board Inlet Temp	23 °C (73.4 °F)	3 °C (37.4 °F)	42 °C (107.6 °F)	-7 °C (19.4 °F)	47 °C (116.6 °F)

System Board Temperature Historical Data

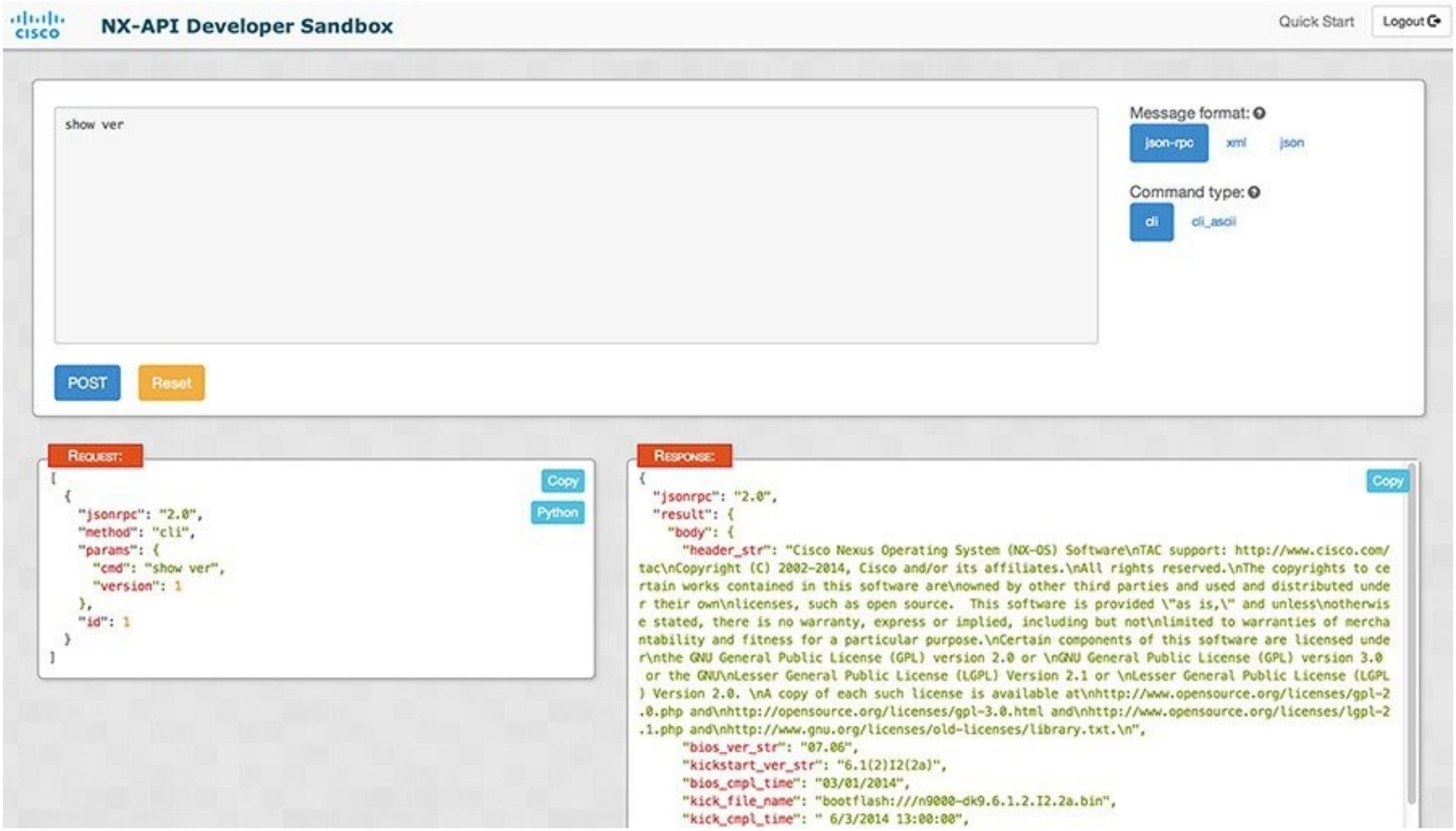
CSV ↗ Export

Total Operation Time	Time Spent in Warning Threshold Range	Time Spent in Critical Threshold Range
72 Days	0%	0%

Supervision et protocoles : les API

- **API : Application Programming Interface**
  - Couche d'abstraction pour simplifier l'administration d'une application ou d'un équipement
  - De plus en plus d'éditeurs et de fabricants proposent une API, mais la standardisation reste compliquée (SOAP, **REST**…)
- Permettent de manipuler des données plus complexes
  - Retours en **XML**, ou en **JSON**, ou autre
- Permettent une gestion plus fine de la **sécurité** (authentification, chiffrement…)

Utiliser une API : GET/POST HTTP



Utiliser une API : Ansible et Cisco

```yaml
- name: Nexus - Configuring spanning tree to pvst
  nxos_config:
    save_when: modified
    provider: "{{ provider }}"
    lines:
      - spanning-tree mode rapid-pvst
    match: line
  when: ('telnet' not in group_names) and
        ('nexus'  in group_names)

- name: Catalyst - Pushing bpdufilter, bpduguard and recovery configuration
  ios_config:
    save_when: never
    authorize: yes
    provider: "{{ provider }}"
    lines:
      - spanning-tree portfast bpdufilter default
      - spanning-tree portfast bpduguard default
      - errdisable recovery interval 30
      - errdisable recovery cause bpduguard
    match: line
  when: ('telnet' not in group_names) and
        ('nexus' not in group_names)
  notify: save

- name: Nexus - Configuring default port to edge port
  nxos_config:
    save_when: modified
    provider: "{{ provider }}"
    lines:
      - spanning-tree port type edge default
    match: line
  when: ('telnet' not in group_names) and
        ('nexus' in group_names)

- name: Nexus - Pushing bpdufilter, bpduguard and recovery configuration
  nxos_config:
    save_when: modified
    provider: "{{ provider }}"
    lines:
      - spanning-tree port type edge bpdufilter default
      - spanning-tree port type edge bpduguard default
      - errdisable recovery interval 30
      - errdisable recovery cause bpduguard
    match: line
  when: ('telnet' not in group_names) and
        ('nexus' in group_names)
```

- API disponible **suivant le modèle**…
- … et **la licence** !
- Dans le cas d'Ansible, certains modules utilisent l'API, sinon il faut se rabattre sur des modules qui exploitent la CLI…

Supervision : petite conclusion

- **Le protocole à utiliser dépend du besoin et du contexte !**
- **Le plus répandu : SNMP**
  - **SNMPv3 encore très rare**, SNMPv1 toujours standard chez beaucoup d'équipementiers…
  - Sur certains équipements (ex : Ubiquiti ?), pourrait commencer à disparaître (pour forcer l'achat de licences pour utiliser les API ?)
  - Nécessite d'avoir les modules adéquats (ex : supervision d'un onduleur industriel)
- **Les API sont prometteuses**
  - Permettent de transmettre **plus de données**, et de **façon structurée** (JSON)
  - Permettent de mieux gérer **l'authentification et le chiffrement**
  - Encore peu disponibles, et nécessitent un serveur Web côté client

Supervision : quelques outils du marché

Commençons par du simple

- **Ping**

```
librenms@librenms:~$ ping 10.0.0.6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=64 time=0.076 ms
^C
--- 10.0.0.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.053/0.064/0.076/0.014 ms
```

- **Telnet**

```
librenms@librenms:~$ telnet 127.0.0.1 80
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
```

- **Curl**

```
librenms@librenms:~$ curl 127.0.0.1 80
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8" />
        <meta http-equiv="refresh" content="0;url='http://127.0.0.1/login'" />

        <title>Redirecting to http://127.0.0.1/login</title>
    </head>
    <body>
        Redirecting to <a href="http://127.0.0.1/login">http://127.0.0.1/login</a>.
    </body>
</html>^C
```

Un outil basique : PHP Server Monitor

SERVER MONITOR | Status | Servers | Log | Users | Config | Update | Help | Welcome, Pep

Servers

SERVERS

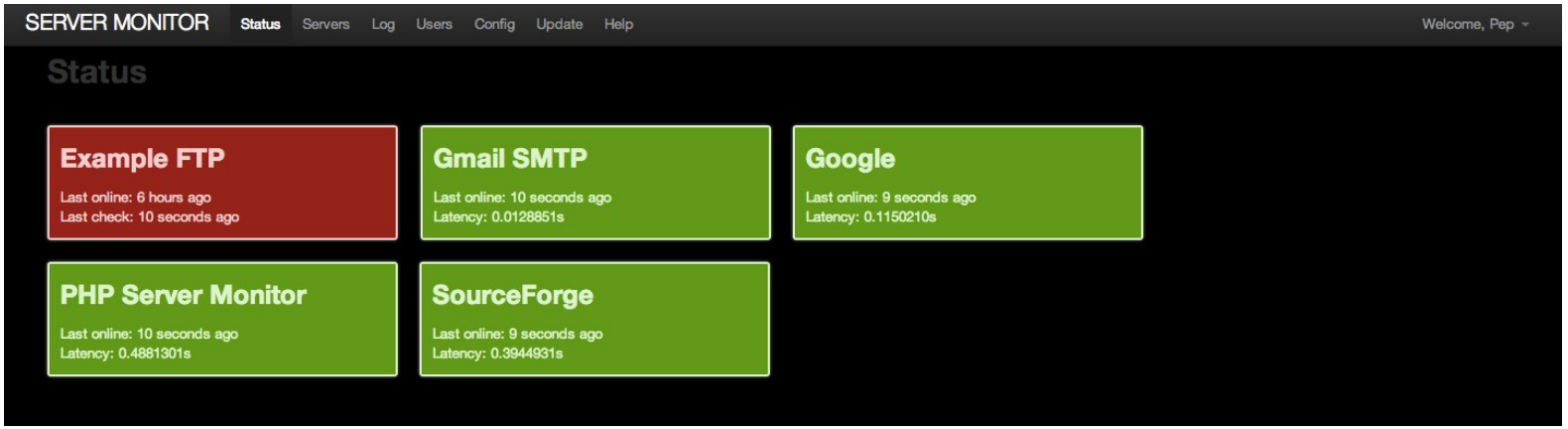
+ Add new

Update

	Label	Domain/IP	Port	Type	Latency	Last online	Monitoring	Send Email	Send SMS	Action
off	Example FTP	ftp.example.org	21	Service	0.0005 s	6 hours ago	Yes	Yes	Yes	
on	Gmail SMTP	smtp.gmail.com	465	Service	0.0129 s	about a minute ago	Yes	Yes	Yes	
on	Google	http://www.google.com	80	Website	0.115 s	about a minute ago	Yes	No	No	
on	PHP Server Monitor	http://www.phpservermonitor.org	80	Website	0.4881 s	about a minute ago	Yes	Yes	Yes	
on	SourceForge	http://sourceforge.net/	0	Website	0.3945 s	about a minute ago	Yes	Yes	Yes	

Powered by PHP Server Monitor v3.0.0.

Back to top



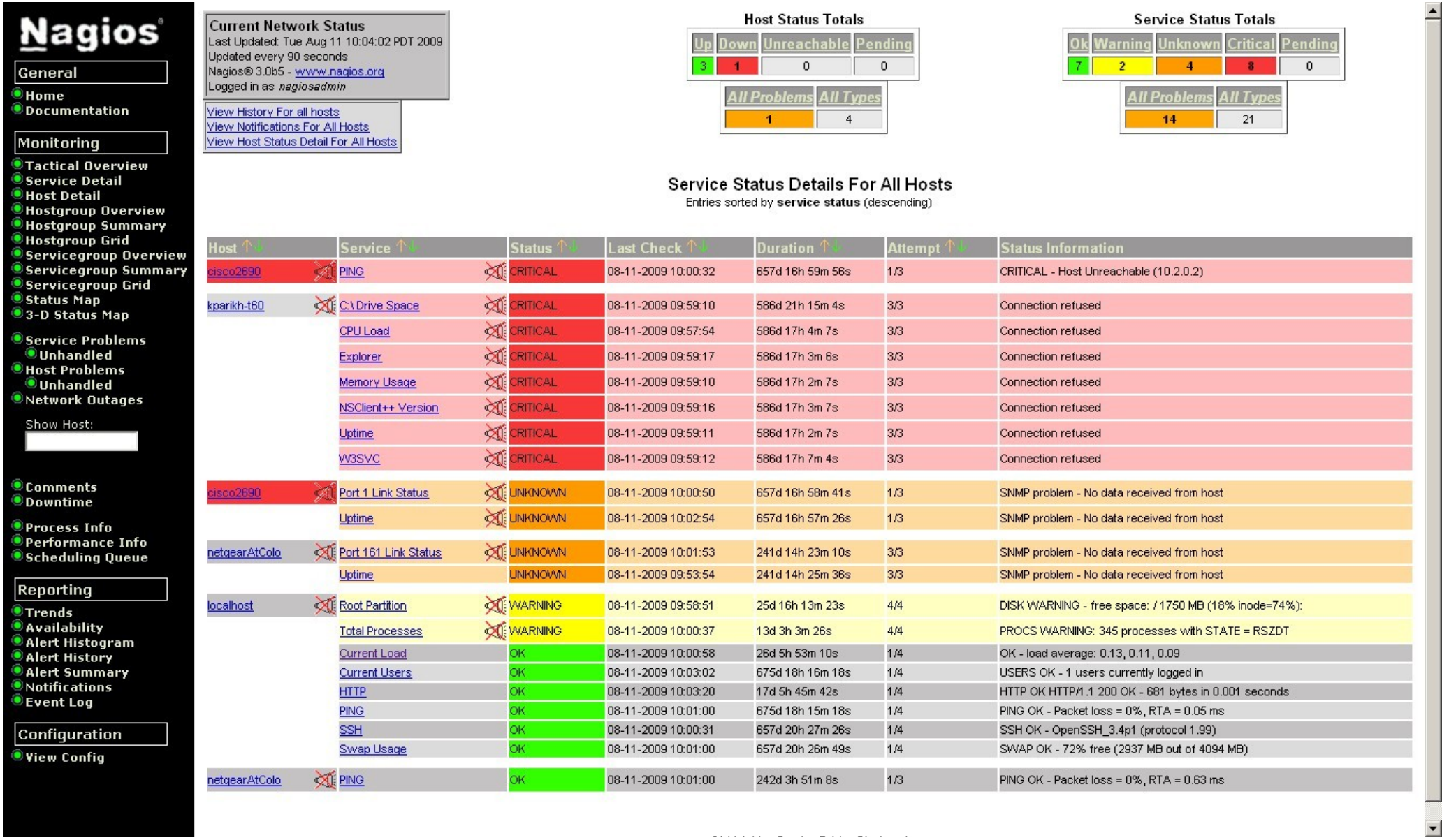
Nagios

- Créé en 2002 (GPLv2)
- S'articule autour :
  - D'un moteur de collecte et d'ordonnancement
  - D'une interface Web
  - De **fichiers de configuration** à plat
  - De **plugins**, pouvant être développés **dans n'importe quel langage** (Bash, Ruby, Python…) et devant juste respecter la norme suivante :
    - **0 : OK** (tout va bien)
    - **1 : WARNING** (le seuil d'alerte est dépassé)
    - **2 : CRITICAL** (le service a un problème)
    - **3 : UNKNOWN** (impossible de connaître l'état du service
- Souvent utilisé avec des extensions comme **NagVis** (créé en 2004) pour avoir un rendu plus visuel

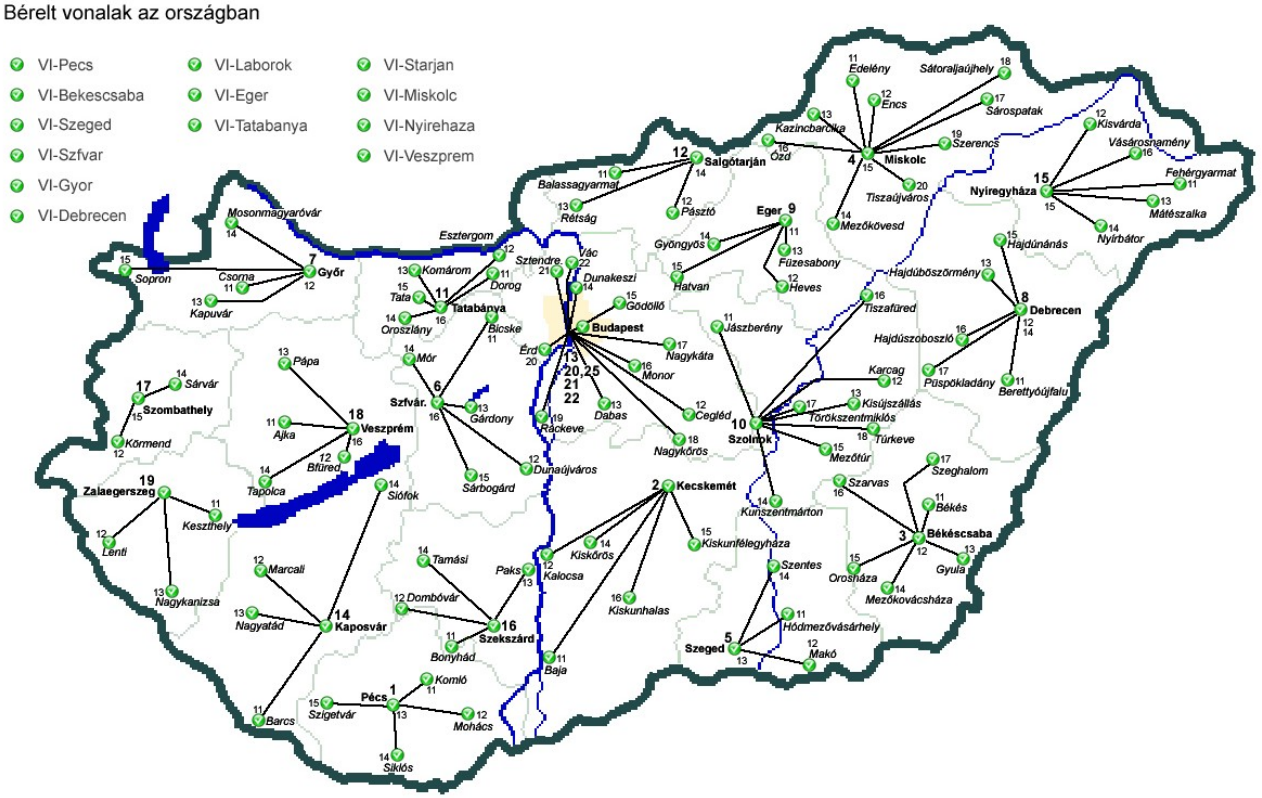
Nagios et son environnement

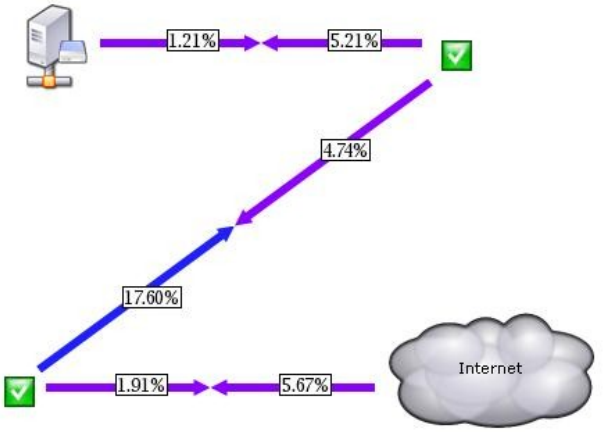
- **NDOUtils : Nagios Data Output Utilities**
  - Permet d'exporter les données de Nagios dans une base MySQL
- **RRDtool : Round-Robin Database Tool**
  - Issu du projet MRTG (Multi Router Traffic Grapher)
  - Permet la gestion de données sous forme de **séries temporelles** (représentant l'évolution d'une donnée dans le temps)
  - Permet la représentation de ces données sous forme graphique
  - Utilisé dans de **très nombreux** outils de supervision
- **NRPE : Nagios Remote Plugin Executor**
  - Permet l'exécution de scripts directement sur les clients
  - Permet l'exécution de scripts vers des clients injoignables (utilisé alors en tant que proxy)

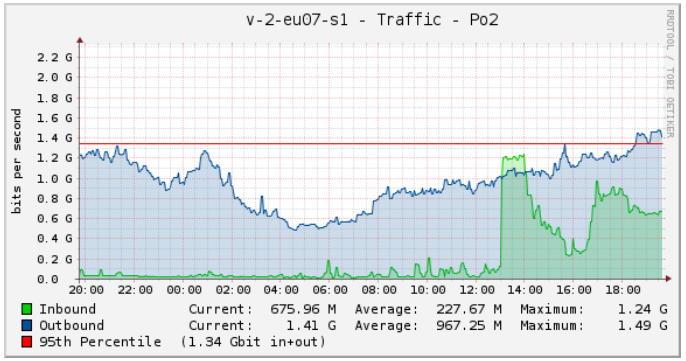
Nagios par l'image

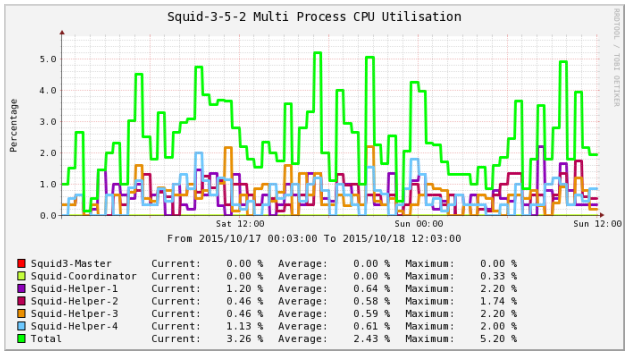


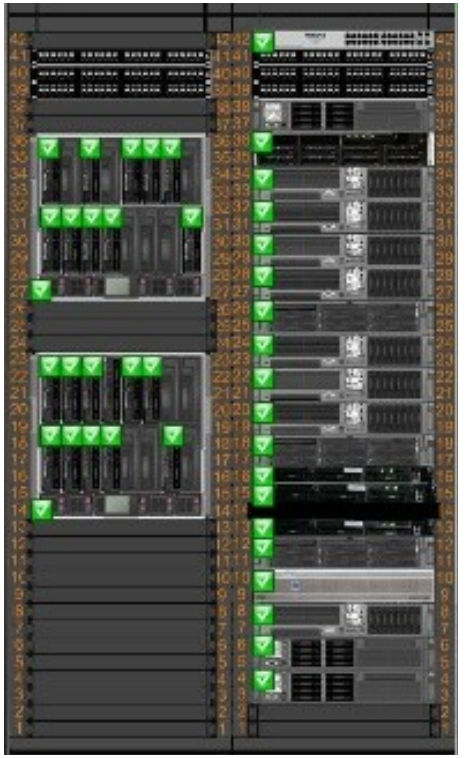
NagVis et RRDtool par l'image











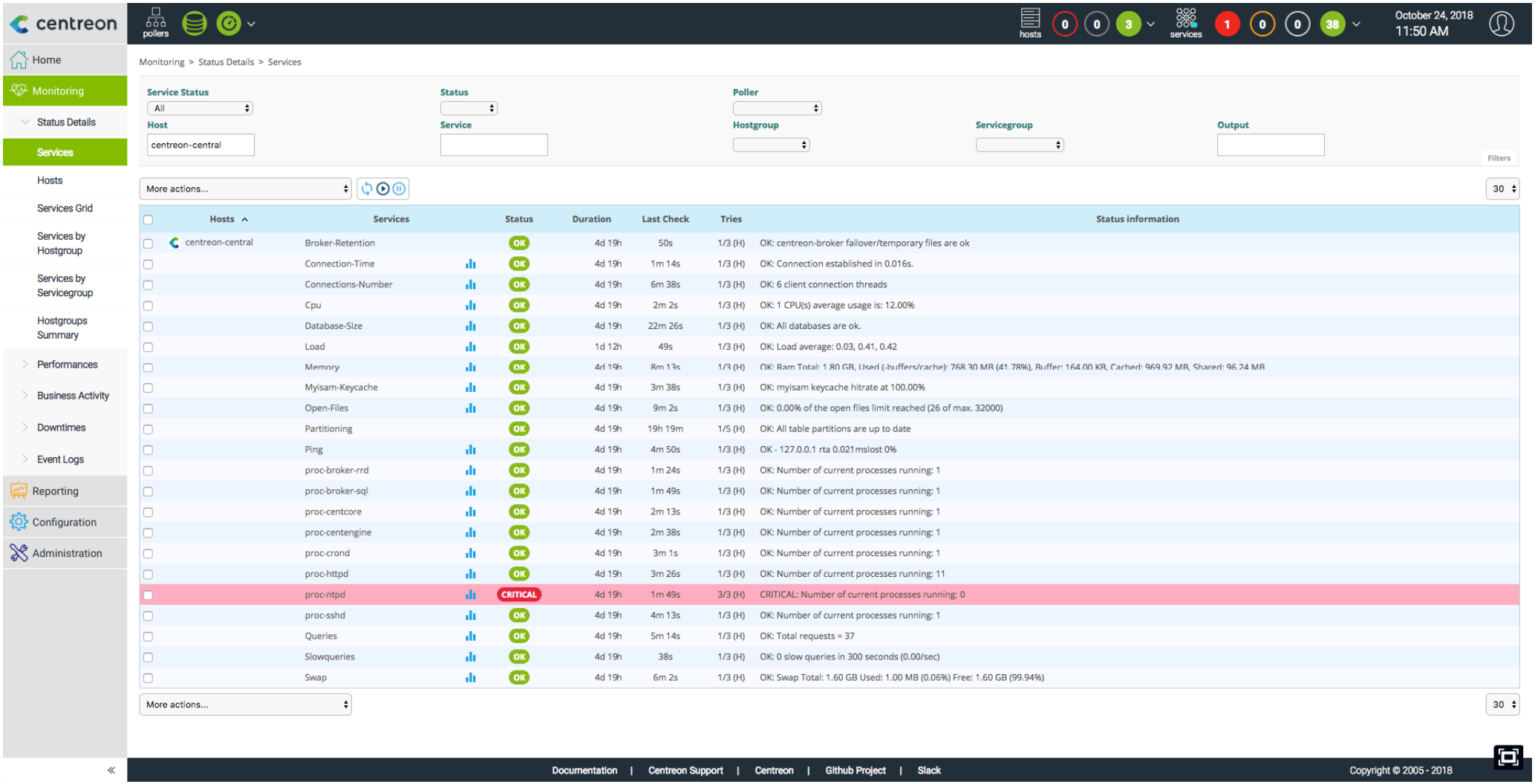
Centreon

- Au début, une **surcouche Web à Nagios** :
  - Meilleure lisibilité des alertes
  - Gestion des configurations plus accessible (directement via la GUI Web)
  - Graphiques de performances

- En 2012, introduction du **centreon-engine** pour s'émanciper complètement de Nagios

- Aujourd'hui, solution complète proposant :
  - Une solution de supervision distribuée : **Centreon Engine** (moteur de collecte), **Centreon Broker** (gestionnaire d'évènements) et **Centreon Web**
  - Des **packs de plugins** payants et gratuits
  - Des outils d'analyses et de reporting (dont des suites payantes)

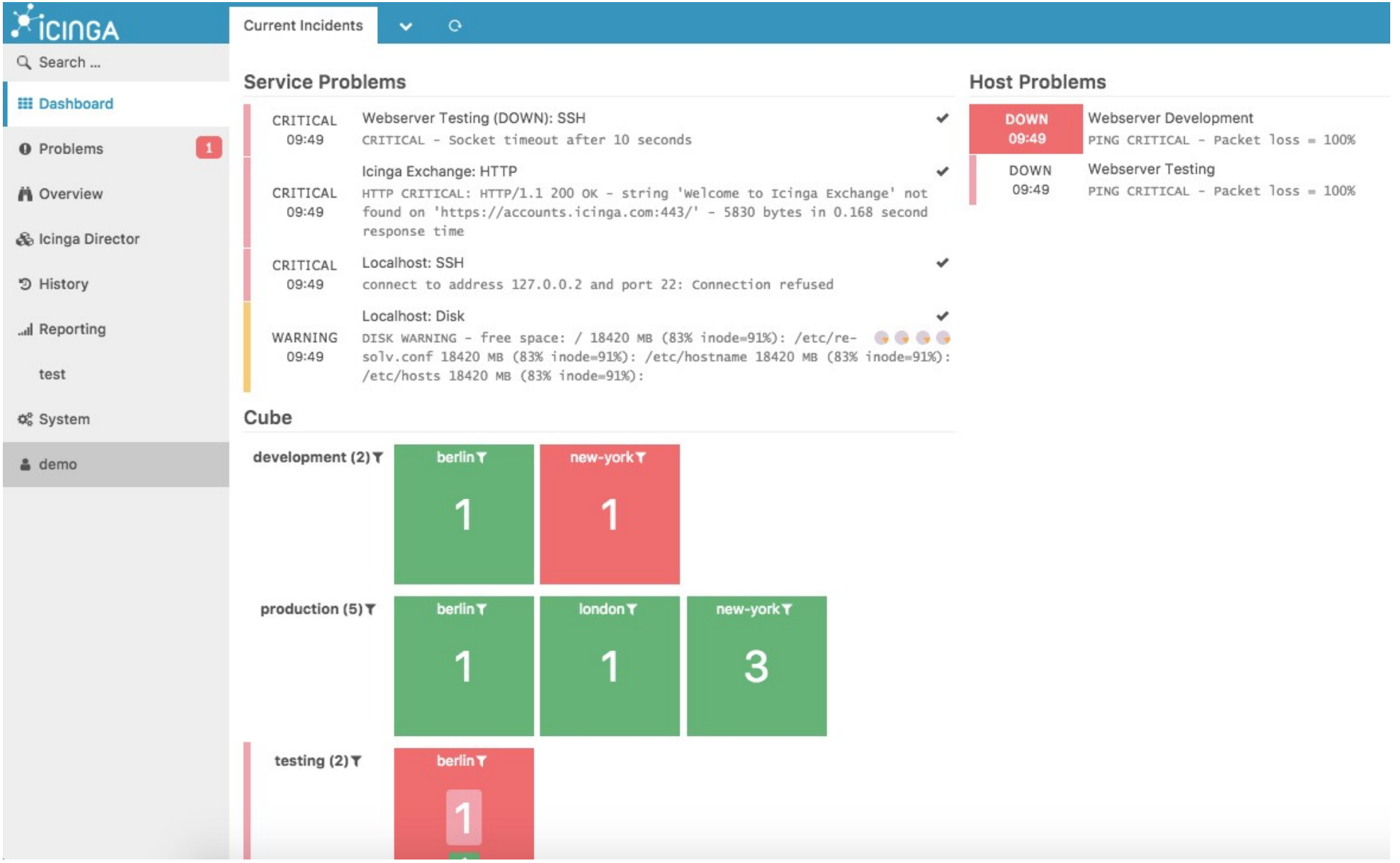
Centreon par l'image



Icinga

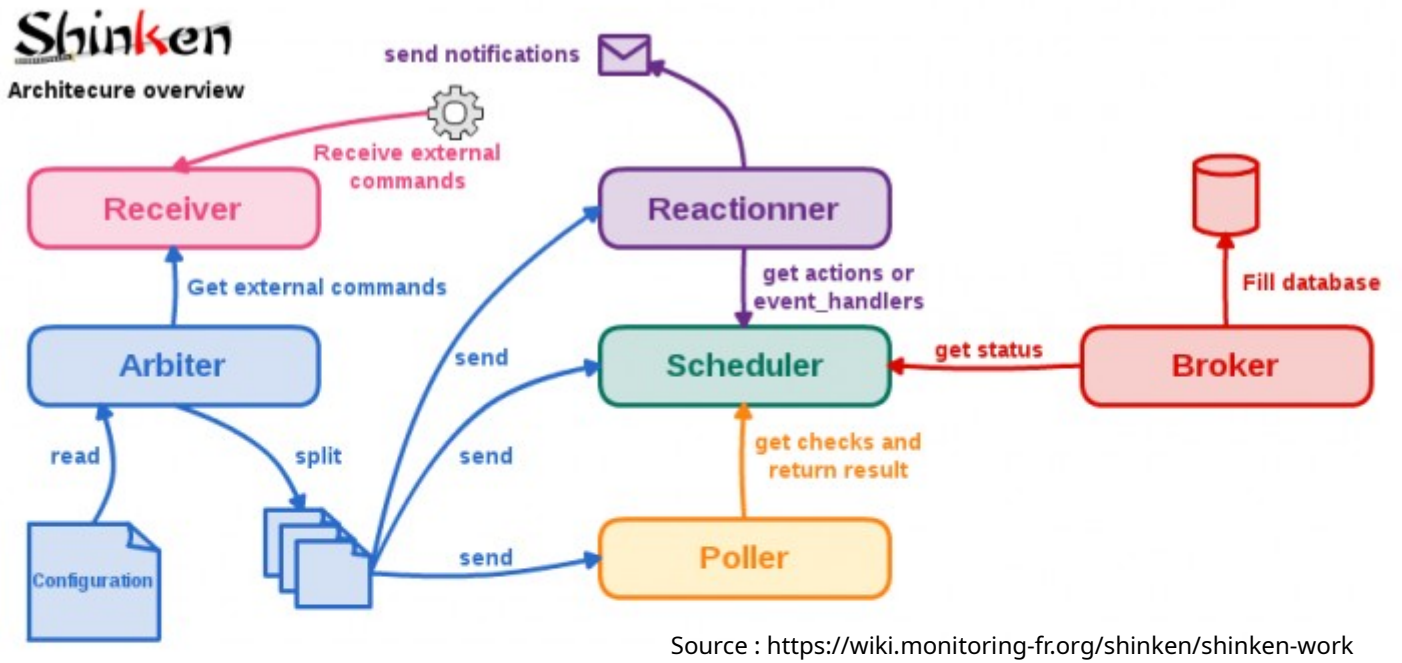
- Fork de Nagios créé en 2009 (GPLv2)
- Né de la volonté de **moderniser** Nagios :
  - **Cycle de développement plus rapide**
  - Interface Web moderne
  - Support des bases de données Oracle et PostgreSQL
  - API REST
- **Maintien de la compatibilité** des configurations et des plugins avec Nagios (attention : Icinga2 ne garantie plus cette compatiblité !)

Icinga par l'image



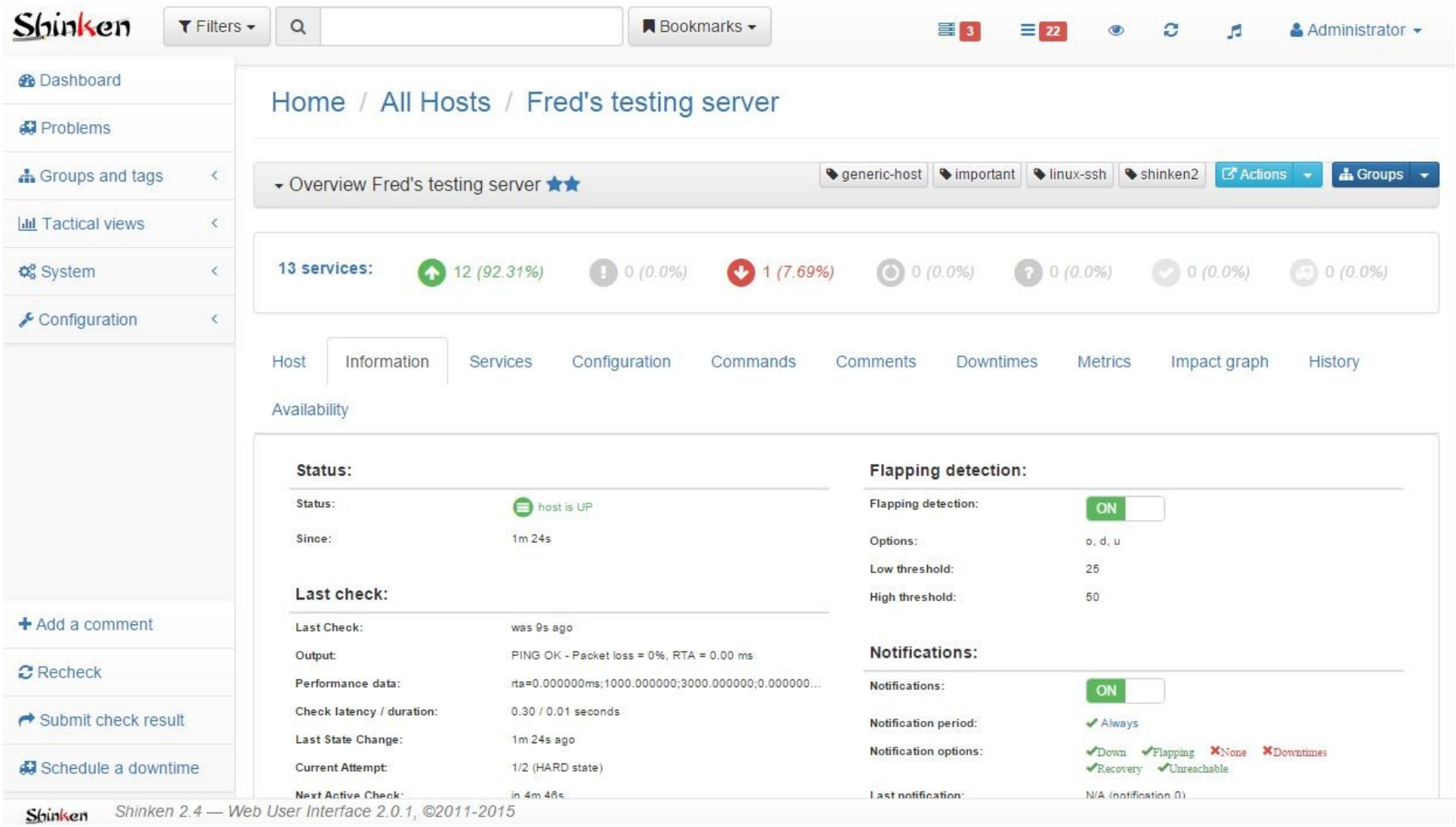
Shinken

- Projet Libre (AGPL) issu d'un POC de refonte de Nagios, publié en 2009
- Développé en **Python**
- Découple les rôles de Nagios



Source : https://wiki.monitoring-fr.org/shinken/shinken-work

Shinken par l'image

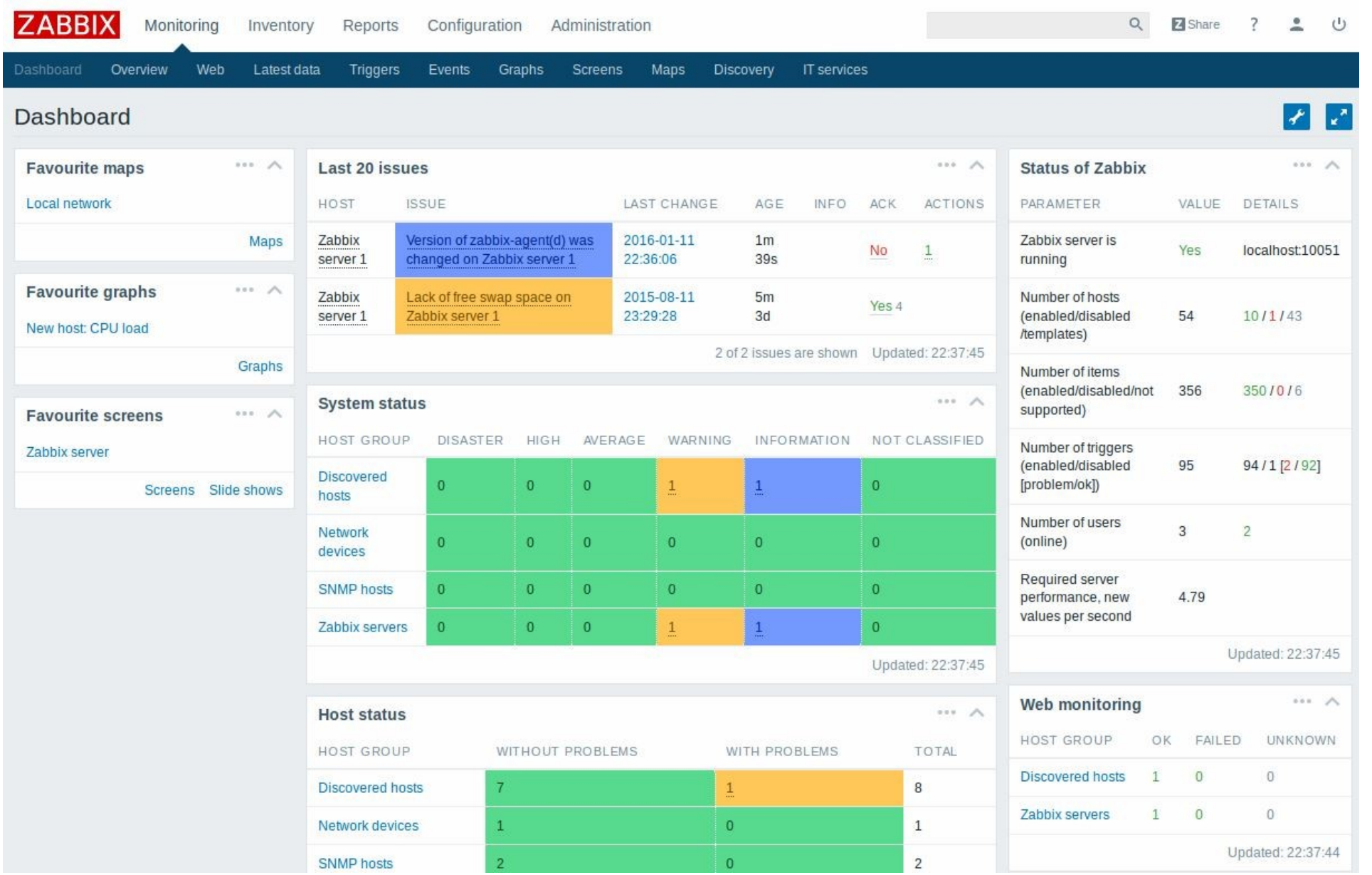


Zabbix

- Projet créé en 1998 et devenu Libre (GPLv2) en 2001
- Peut de faire de l'auto-discovery
- Pensé pour la performance, théoriquement capable de superviser des centaines de milliers d'assets

- S'articule autour des composants :
  - **Zabbix Server** : collecte les données et gère les alertes
  - **Zabbix Agent** : collecte locale des données d'un système
  - **Zabbix Frontend** : l'interface Web centralisée
  - **Zabbix Proxy** : permet d'atteindre des systèmes éloignés

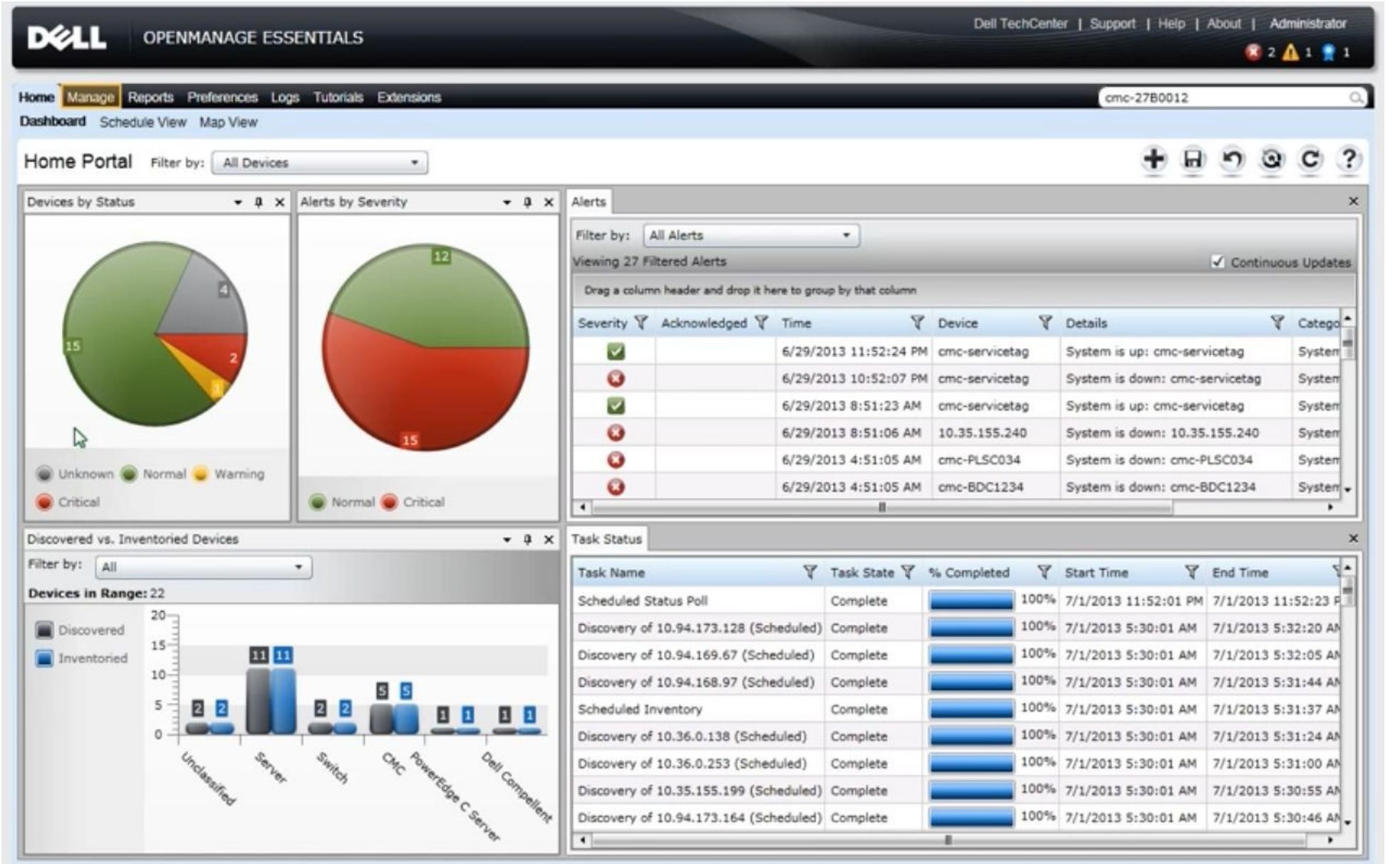
Zabbix par l'image



Dell OpenManage Essentials

- Console d'administration et de supervision d'un parc matériel (Dell et autres vendeurs)
- S'interface avec les équipements via **IPMI**, **SNMP**, **WS-MAN** (Web-Services Management)…
- Permet de :
  - Déployer des **templates de configuration** sur les serveurs (BIOS, iDRAC…)
  - Gérer les **mises à jours de firmwares**
  - Surveiller **l'état des composants** et **lever des alertes** (tickets automatiques au support Dell)
- Peut faire de l'auto-discovery

Dell OME par l'image

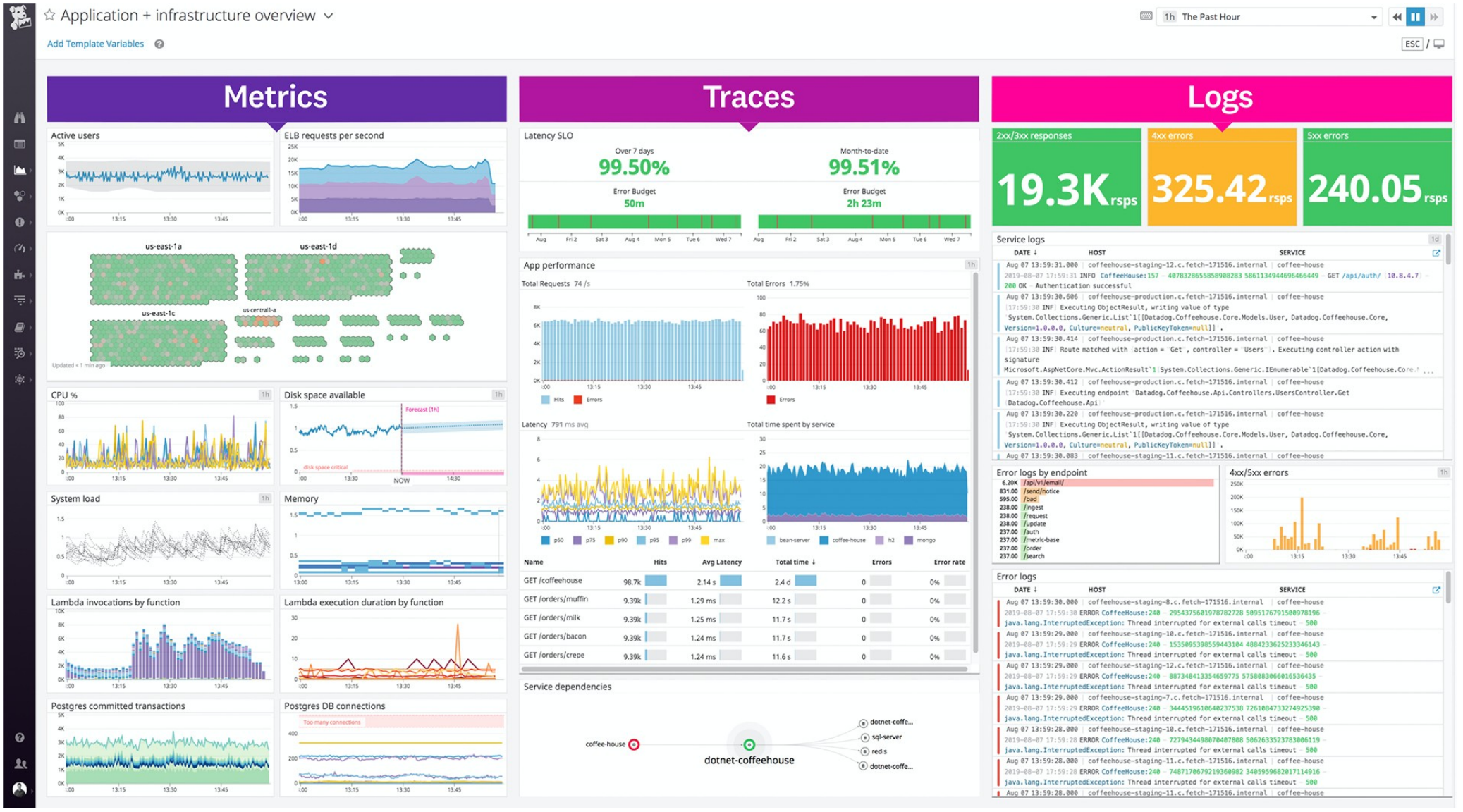


La supervision-as-a-service

- Aujourd'hui, plusieurs solutions « Cloud » existent : ☁️
  - Des offres issues des éditeurs historiques (comme Centreon)
  - Des nouveaux arrivants, par exemple Datadog (fondée en 2010 par 2 français)

- Avantages :
  - Un déploiement **simplifié** et une **rationalisation** des solutions de supervision et de métrologie
  - Un tarif adapté à la quantité de sondes mises en place
  - Une offre de **support** adaptée
- Inconvénients :
  - Un risque de **sécurité** important à prendre en compte
  - Un **risque financier** si l'évolution de

Datadog par l'image



Les outils : petite conclusion

- Une **très grande** variété d'outils disponibles
- Quelques exemples de critères de sélection :
  - **Le cahier des charges !**
  - Le **type d'assets** à superviser
  - L'environnement existant
  - La pérennité de l'outil (communauté, stabilité, support…)

- Quelques exemples de **mauvais** critères de sélection : ⚠️
  - Les **affinités** (ou non) avec le socle technique
  - Le **coût**
  - Les compétences actuelles des équipes

La métrologie

La métrologie : pourquoi ?

Définition

*« **La métrologie est la science de la mesure.*** 

On peut la découper en trois composantes :

- *La **définition** des unités de mesures*
- *La **réalisation** des mesures*
- *La **traçabilité** et **l'exploitation** des mesures dans le temps »*

Source : Wikipédia

Pourquoi relever des métriques ?

- Pour faire des **statistiques** :
  - Mesure des **taux de disponibilité** (SLA)
  - Surveiller les **performances** d'un environnement dans le temps
  - **Comparer** des environnements (ex : avant/après mise à jour)
  - **Étalonner** son infrastructure
  - **Anticiper** les besoins d'évolutions

- Pour **identifier des problèmes** :
  - **Incidents récurrents = problème sous-jacent !** 
  - Pertes de performances difficilement perceptibles pour la supervision
  - Tentatives d'intrusion ou d'attaques dissimulées
  - Fuites de données

La métrologie : comment ?

Comment relever des métriques ?

- Processus simplifié :
  - 1 : Définir ce qu'on souhaite mesurer et **pourquoi**
  - 2 : Sélectionner l'outil **adéquat** (faire des POC !), penser **« scalabilité »**
  - 3 : Commencer à injecter les métriques, suivre le dimensionnement de l'infrastructure
  - 4 : Commencer à exploiter les métriques, créer les premiers dashboards

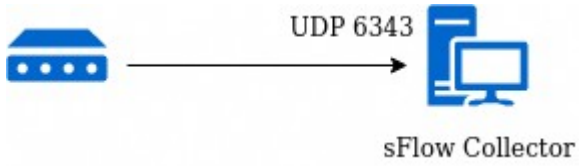
- **Pas nécessairement besoin de « temps réel »**, on pourra opérer des collectes régulières (dépend du contexte)
- 2 approches opposées qui reviennent souvent :
  - Collecter **trop de données** (mais ne pas savoir quoi en faire)
  - Collecter **juste le minimum vital** (mais risquer d'en manquer)

Les protocoles 2 : le retour 😓

sFlow

- Protocole industriel standard
- Fonctionne sur la base d'**échantillons**, centré sur la **couche 2 OSI**
- **Pas de notion de flux** au sens « suite de packets »

- 3 notions :
  - ***flow*** : échantillon de 1/n packets
  - ***counter*** : compteur d'interface
  - ***sFlow datagram*** : échantillons et compteurs envoyés en UDP à un **sFlow Collector**



NetFlow

- Protocole développé par Cisco (supporté par d'autres fabricants, mais souvent sous un autre nom…)
- Fonctionne sur la base de **flux réseaux (couches 4 et 5 OSI)**

- 3 notions :
  - ***NetFlow Exporter*** : agrège les flux
  - ***NetFlow Collector*** : collecte et met en forme les flux
  - ***Analysis Application*** : analyse les flux selon des filtres définis (détection d'attaques par déni de service, de tentatives d'intrusion…)

- Un flux ~= une suite unidirectionnelle de paquets entrant sur la même interface, possédant les mêmes IP et ports sources et destination

RMON et SNMP (encore lui !)

- **RMON : Remote Network Monitoring**
- Protocole industriel standard
- S’appuie sur **SNMP** et un modèle client-serveur

- 2 versions :
  - **RMON 1** : centré sur la **couche 2 OSI**
  - **RMON 2** : prend en charge les **couches 4 et 5 OSI**

- Une **sonde** ou agent RMON
  - **collecte** et applique un pré-traitement aux données
  - peut être intégrée dans un équipement réseau, installée sur un serveur ou déployée sous forme d’appliance physique

Exemples de statistiques RMON

- En RMON 1 :
  - Quantité de paquets reçus sur un segment
  - Quantité de paquets de broadcast
  - Quantité de paquets échangés entre deux addresses MAC
  - Addresses MAC les plus bavardes

- En RMON 2 :
  - Quantité de paquets émis ou reçus par une adresse IP
  - Quantité de paquets reçus pour une adresse IP et un port TCP particuliers
  - …

De nouvelles approches applicatives

- Les infrastructures évoluent : **Cloud Public**, **conteneurs**, **Continuous Delivery**…
- Les approches traditionnelles côté supervision/métrologie sont **trop « statiques »**, il est nécessaire d'être beaucoup **plus souple et réactif**

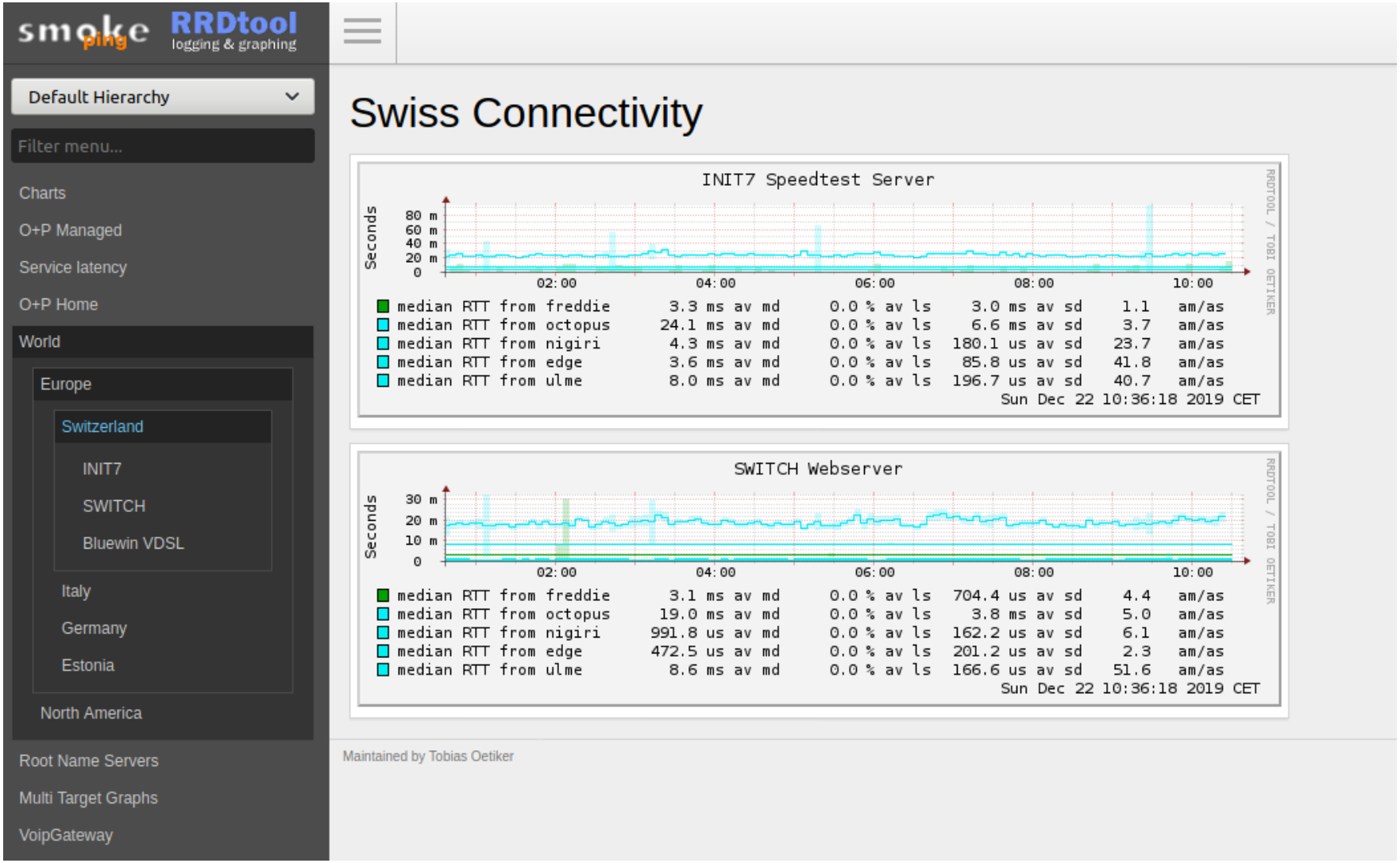
- Apparition d'outils spécifiques, qui peu à peu s'imposent comme une nouvelle norme :
  - Un agent très léger et sans dépendances est déployé avec les VM/conteneurs et envoie (ou met à disposition) ses métriques
  - Un serveur central trie ces métriques par type, tag, provenance…

Métrologie : quelques outils du marché

SmokePing

- Outil d’analyse de **latences**
- Dispose de nombreuses sondes :
  - ICMP
  - DNS
  - SSH
  - LDAP
  - Radius…
- Permet de définir des seuils d’alerte

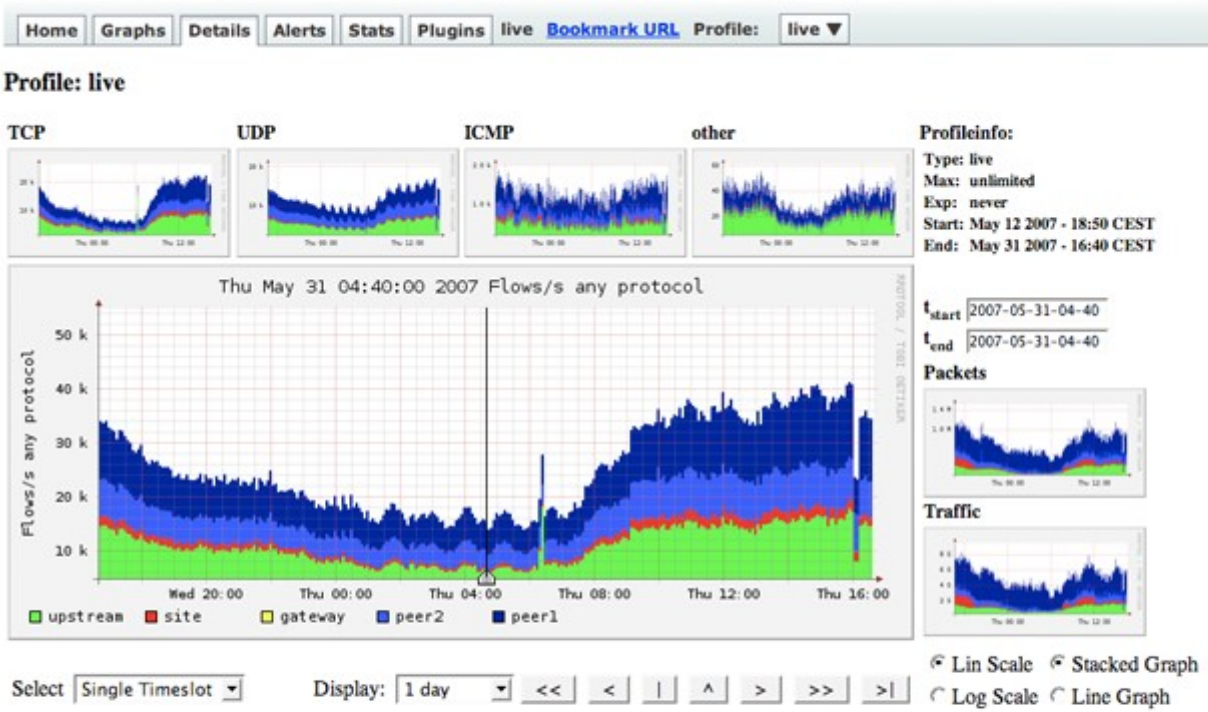
SmokePing par l'image



NfSen

- NfDump collecte les datas NetFlow, mais leur exploitation se fait en CLI
- NfSen est une GUI Web permettant de générer des graphs RRD à partir de ces datas
- Travaille par défaut sur des périodes de 5 min (**≠temps réel !**)
- Permet de travailler en modes :
  - **Flows** (quantité de flux par seconde)
  - **Packets** (quantité de packets par seconde)
  - **Bytes** (quantité d'octets par seconde)
- Permet de définir des seuils d'alerte

NfSen par l'image



Statistics timeslot May 31 2007 - 04:40

Channel:	Flows:	Packets:					Traffic:				
	all:	all:	tcp:	udp:	icmp:	other:	all:	tcp:	udp:	icmp:	other:
peer1	5.2 k/s	423.3 k/s	413.7 k/s	7.3 k/s	1.1 k/s	1.2 k/s	3.4 Gb/s	3.3 Gb/s	21.1 Mb/s	775.2 kb/s	6.7 Mb/s
peer2	3.3 k/s	76.2 k/s	66.9 k/s	7.0 k/s	621.0 /s	1.7 k/s	484.6 Mb/s	459.9 Mb/s	12.5 Mb/s	437.3 kb/s	11.7 Mb/s
gateway	1.0 /s	651.0 /s	600.8 /s	46.6 /s	0 /s	3.7 /s	6.2 Mb/s	6.1 Mb/s	36.4 kb/s	0 b/s	4.4 kb/s
site	467.1 /s	8.9 k/s	6.1 k/s	2.0 k/s	181.7 /s	613.3 /s	38.8 Mb/s	28.3 Mb/s	7.4 Mb/s	104.0 kb/s	2.9 Mb/s
upstream	6.4 k/s	94.2 k/s	84.3 k/s	8.2 k/s	896.4 /s	766.7 /s	588.4 Mb/s	568.2 Mb/s	16.7 Mb/s	685.1 kb/s	2.8 Mb/s

All None Display: Sum Rate

Cacti

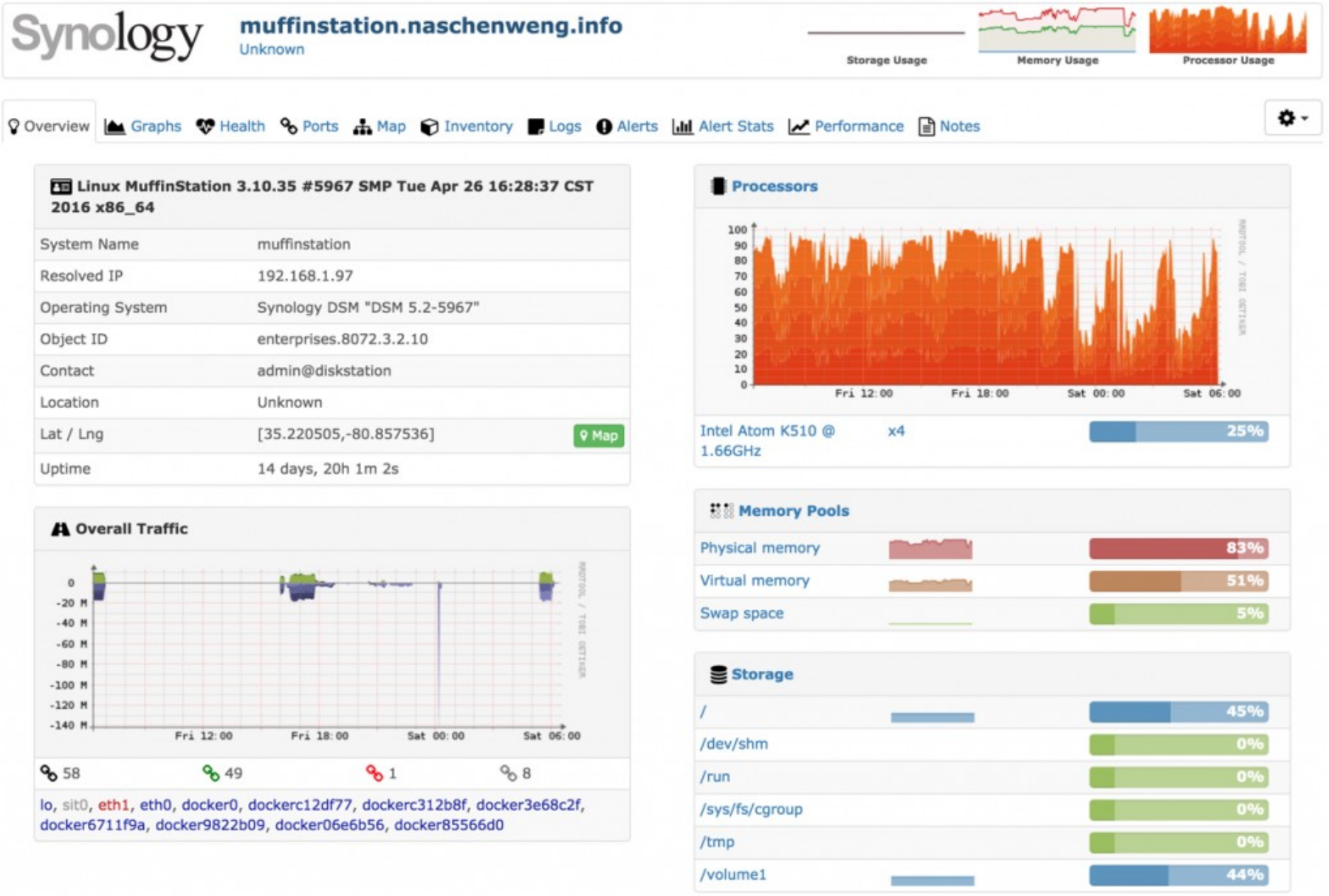
- Pensé comme une interface Web pour **RRDtool**
- Capable de découvrir toutes les interfaces à surveiller sur un équipement réseau
- Capable de collecter d'autres type de métriques (espace disque, CPU...)
- Facilement scalable
- S'appuie sur **SNMP**
- Permet d'analyser des comportements sur des **périodes de temps données**

Cacti par l'image

Observium / LibreNMS

- LibreNMS est un fork d'Observium, davantage tourné vers la communauté
- Permet de faire de l'**auto-discovery**, identifie les métriques à collecter
- **Compatible avec des dizaines d'équipementiers !**
- Dispose d'un système de plugins pour collecter des données « internes » : tables ARP, routes BGP, VLANs configurés…
- Permet de faire du **« billing »** (par quota ou 95th percentile)
- Peut s'interfacer avec des outils de NetFlow/sFlow, de sauvegarde des équipements (Rancid, Oxidized), de syslog…
- Permet de définir des seuils d'alerte

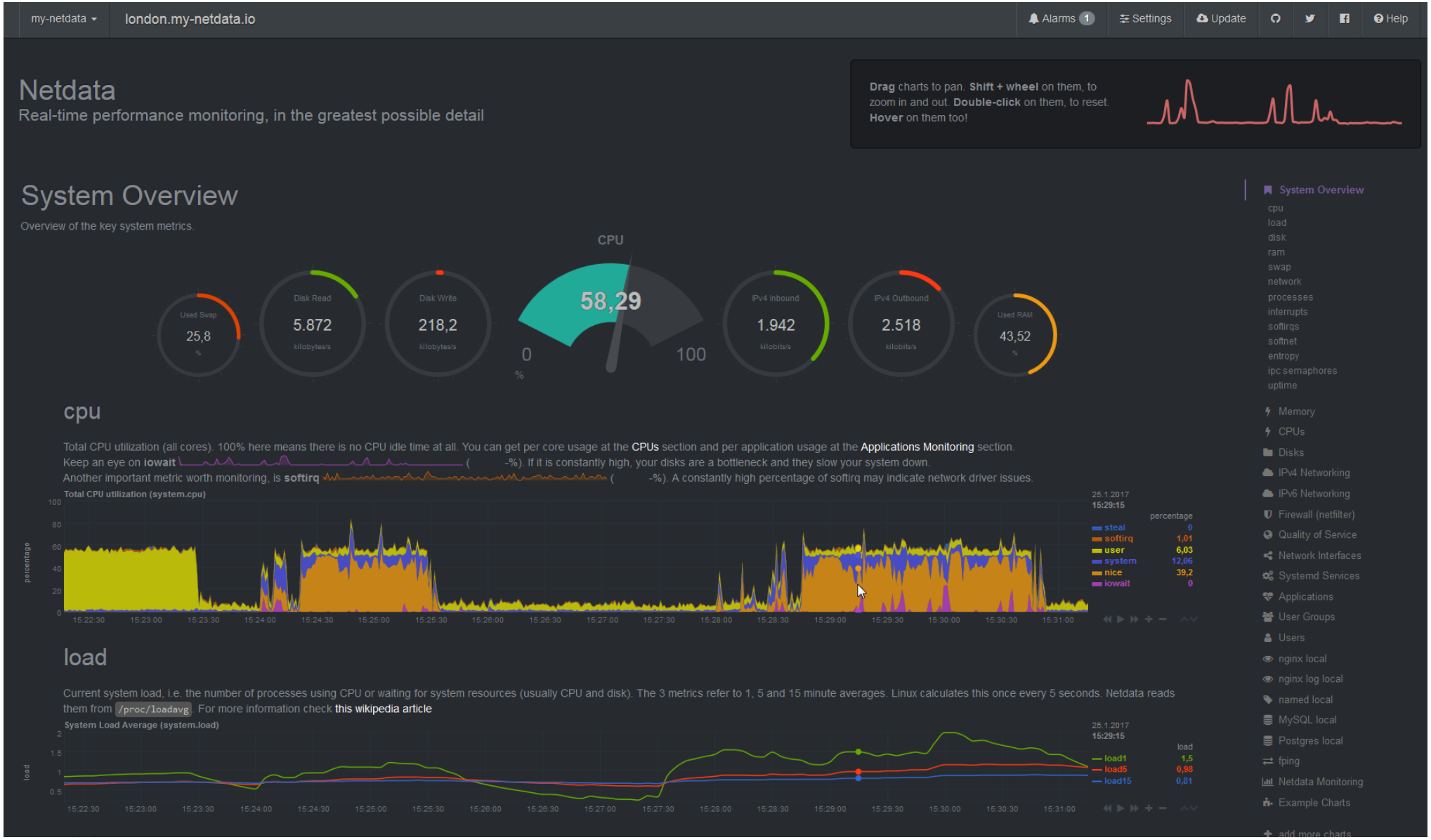
LibreNMS par l'image



Netdata

- Outil de collecte de données installé en local
- **Analyse l'environnement** pour déterminer les métriques à collecter (système, applicatifs…)
- Pensé pour être **léger**, collecte les données de façon **non-intrusive** :
  - Utilise les cycles CPU libres
  - Lit le maximum d'informations dans **/proc**
  - Écrit dans ses logs uniquement à l'arrêt
  - **Aucune dépendance**
- Fournit une interface Web (**à protéger !**)
- **Pas de centralisation** des données collectées, même si l'interface Web permet d'agréger les visualisations de plusieurs machines

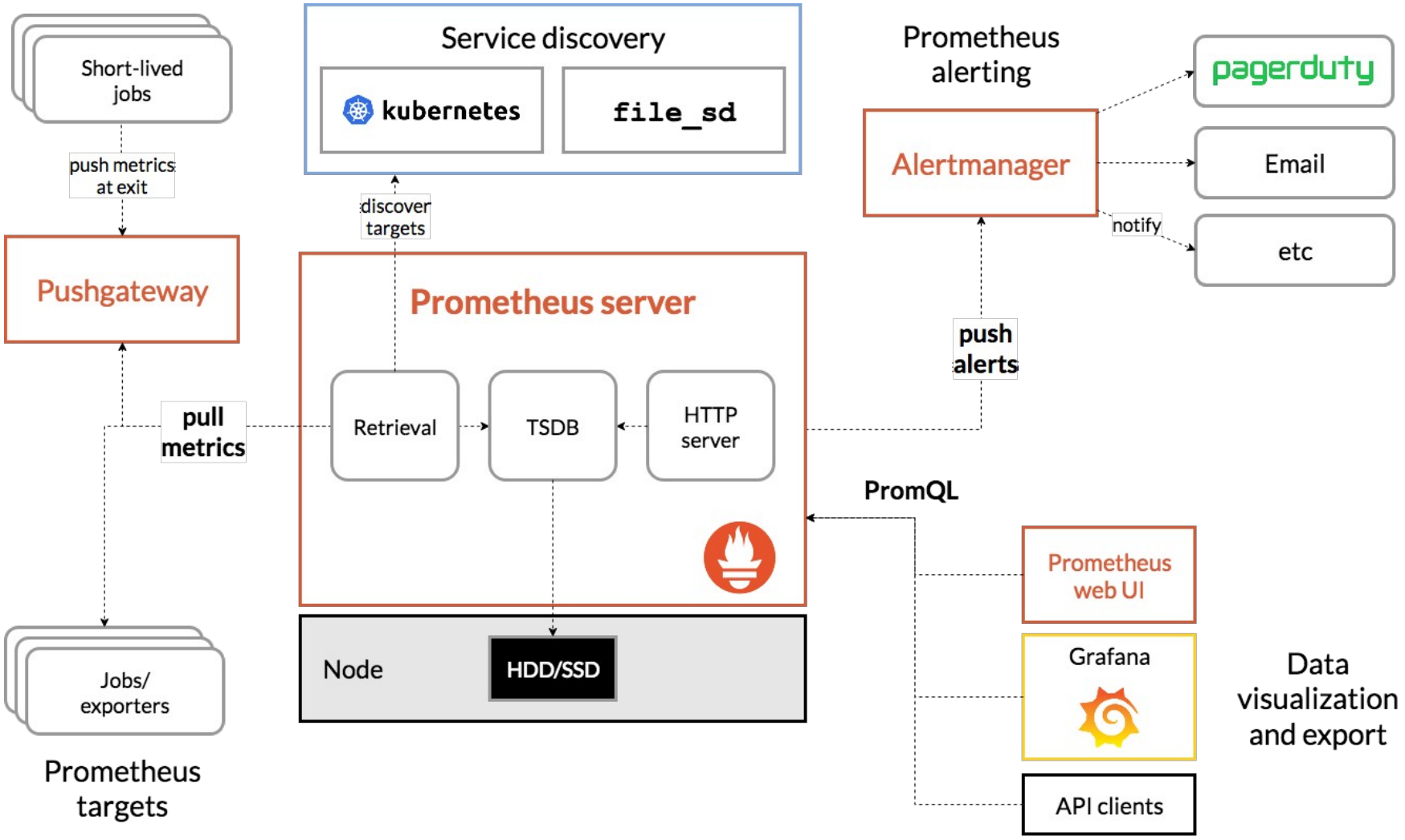
Netdata par l'image



Prometheus

- Projet open-source apparu en 2012
- Développé en Go, **aucune dépendance**
- Modèle de données multidimensionnel : **timeseries et clé/valeur**
- Propose un langage de requête puissant
- Repose sur un modèle **client-serveur** :
  - Les clients mettent les données à disposition via des **exporteurs**, chaque application possédant son propre exporter
  - Le serveur Prometheus vient collecter les données en HTTP régulièrement
- Principalement utilisé en combinaison avec **Grafana** pour visualiser les données

Fonctionnement de Prometheus



Prometheus et Grafana par l'image

Annexe 1 : la supervision des logs

Pourquoi ? Comment ?

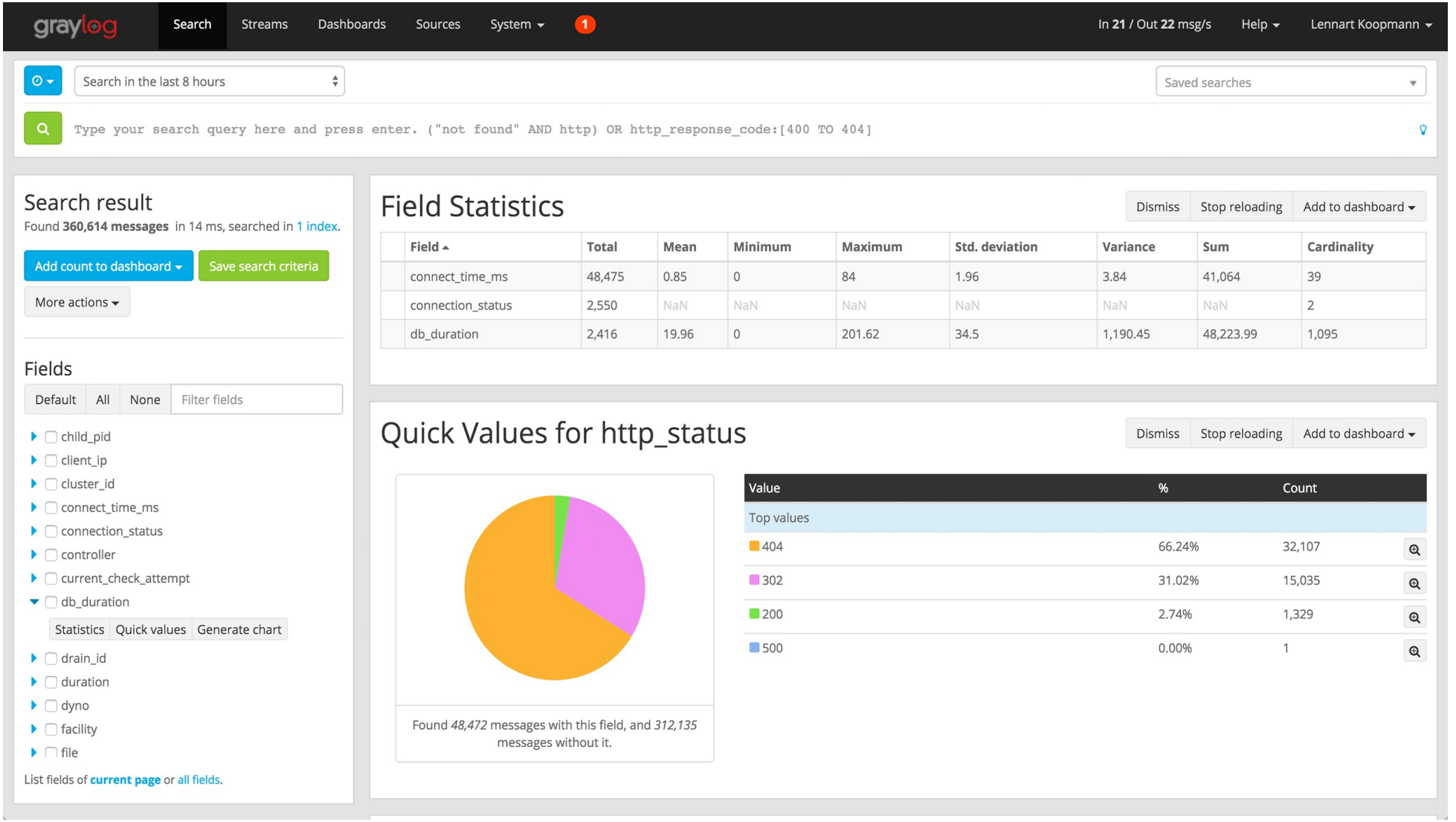
- **Pourquoi ?**
  - Pour des raisons de **sécurité**
    - **tentatives d'intrusions**
    - authentifications échouées
    - tentatives d'accès à des pages normalement inaccessibles
  - Pour détecter des **comportements anormaux** dans les services
    - pages ou scripts en **erreur**
    - **temps d'accès** trop importants…

- **Comment ?**
  - **Centralisation** et **archivage** (raisons légales ou pratiques)
  - Puis **analyse** (expressions régulières, indexation…)

Exemple d’outil 1 : Graylog

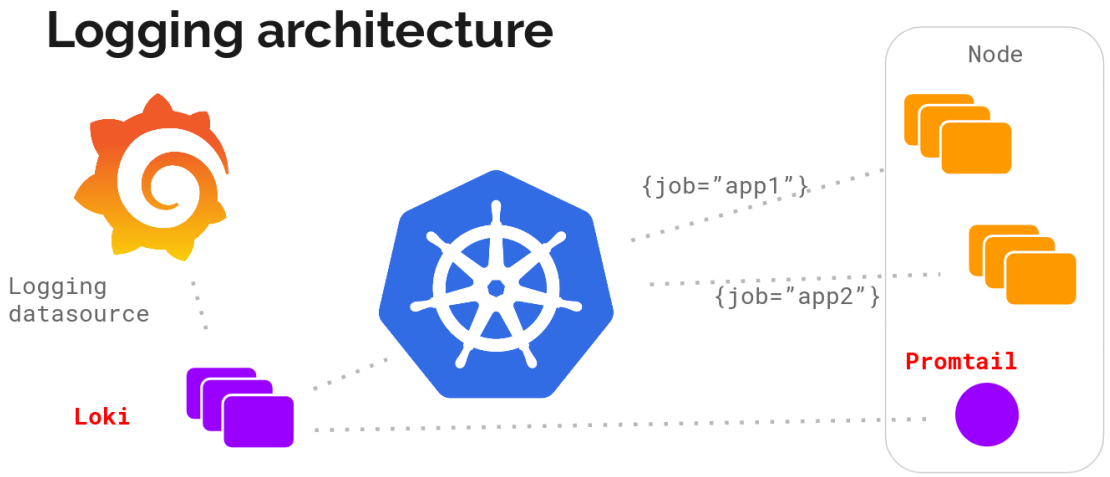
- Outil de centralisation et d’analyse des logs créé en 2009
- S’appuie sur **Elasticsearch** (moteur de stockage, de recherche et d’analyse full-text puissant, multi-tenant et distribué)
- Permet de créer des dashboards pour suivre les logs de son infrastructure
- Permet de définir des seuils d’alerte

Graylog par l'image

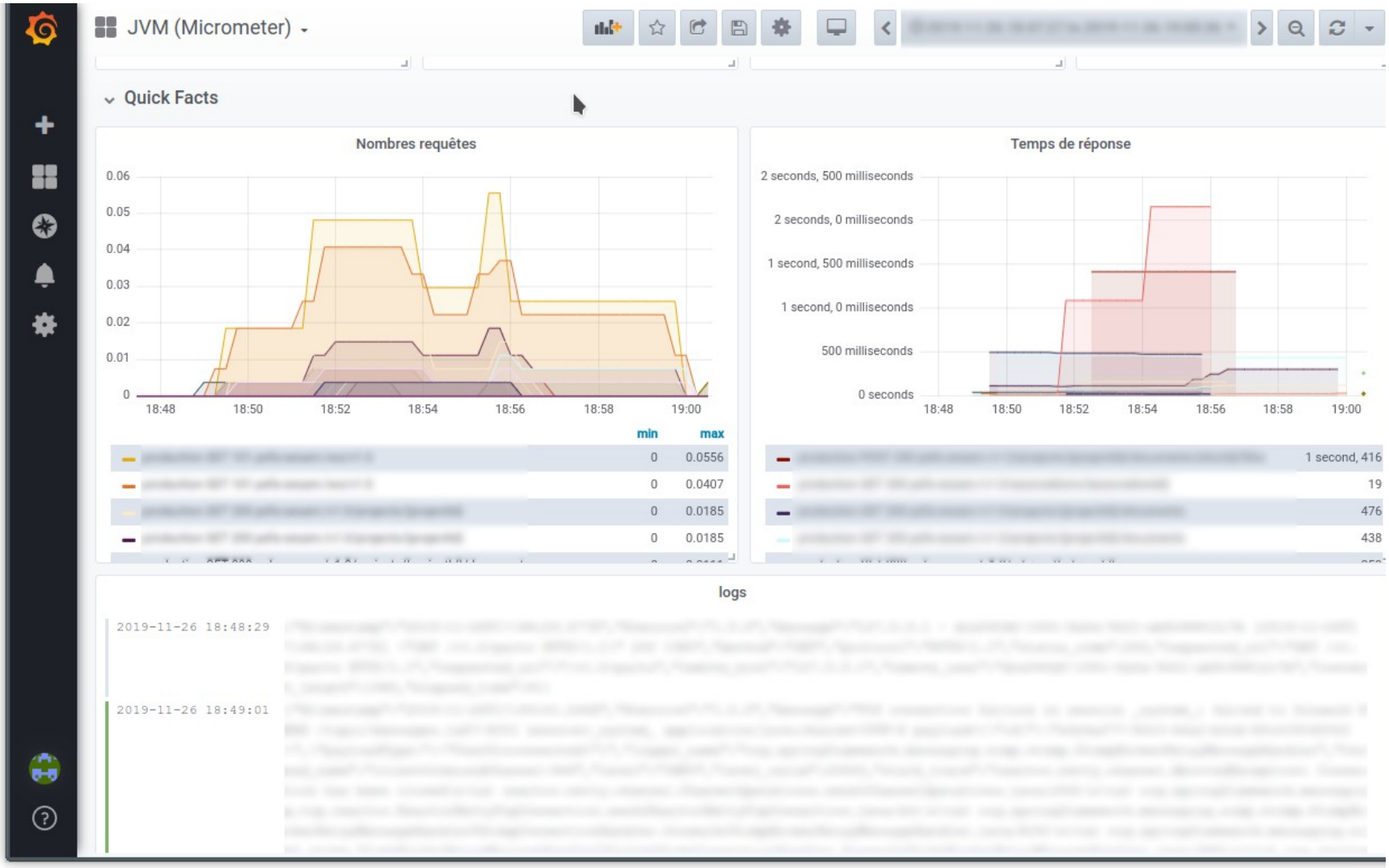


Exemple d’outil 2 : Loki

- Loki = Prometheus, mais pour les logs
- Pour le stockage des données, on reprend l’idée de **timeseries** auxquelles on ajoute des **labels** (pas d’indexation et de structuration des logs, juste des metadatas)
- On s’appuie une fois encore sur **Grafana** pour réaliser des dashboards



Loki et Grafana par l'image



Annexe 2 : aux frontières de la sécurité

Les firewalls

- Un firewall n’a pas qu’un rôle de filtrage : il doit aussi être surveillé
- Plusieurs informations peuvent avoir de l’importance :
  - Les **règles qui ont bloqué des flux** : un « match » n’est pas significatif, mais une série importante dans un cours laps de temps peut indiquer une anomalie
  - Le nombre de **« stateful connections »** actuellement ouvertes : un firewall a des capacités limitées, si on s’approche trop du seuil on prend le risque d’avoir des connexions interrompues ou coupées, ou des dégradations de performances
  - Dans le cas de règles de **QOS**, de **« queuing »**/**« shaping »**, on peut vouloir alerter le client qu’il atteint les limites prévues contractuellement

Le cas des IDS (et IPS)

- **IDS : Intrusion Detection System**
- **IPS : Intrusion Prevention System**

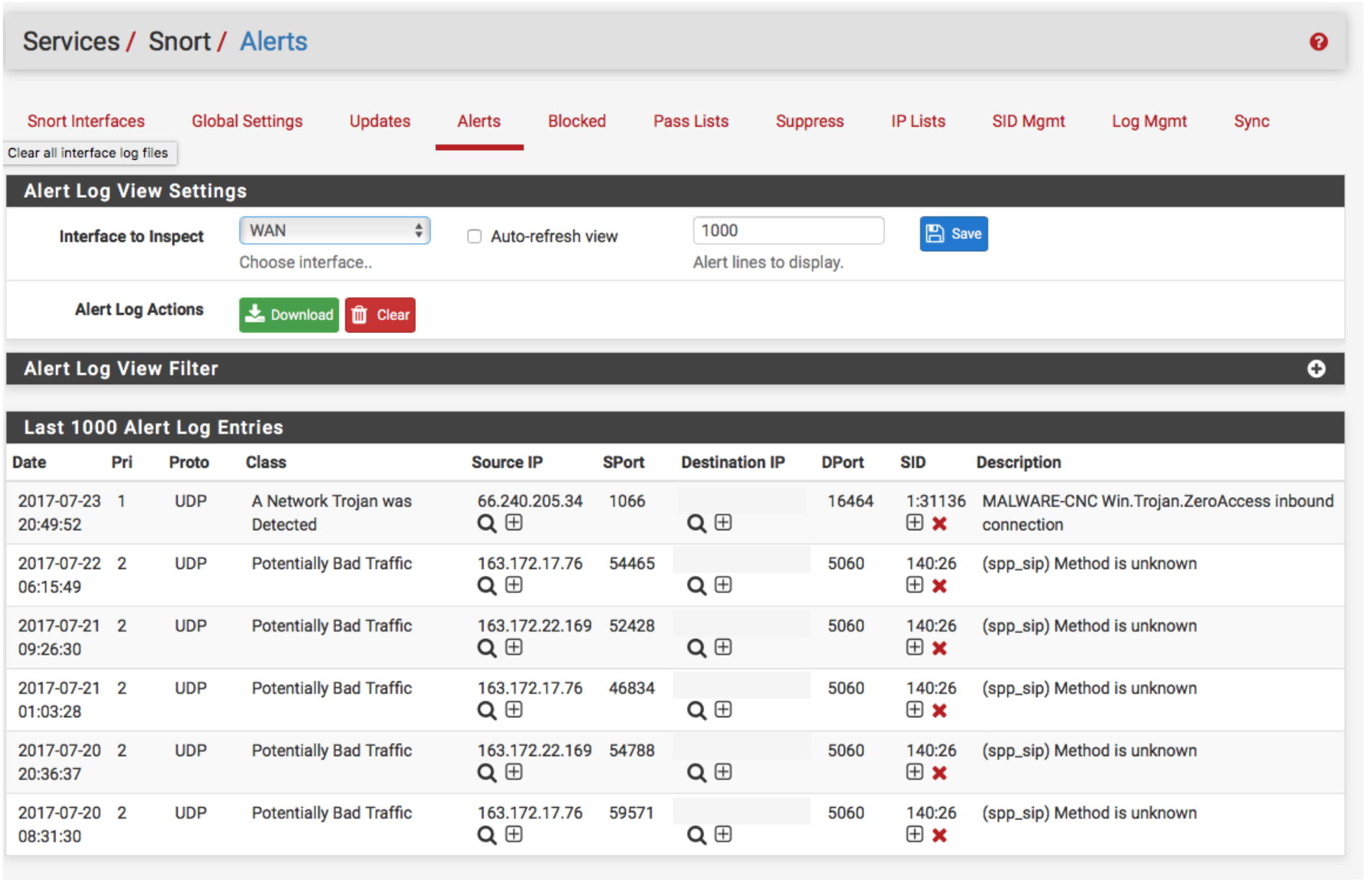
- Peuvent nécessiter des **ressources importantes** (analyse du trafic en temps réel)
- **A ne pas confondre avec des pare-feux**, même si ces derniers incluent parfois des IDS/IPS

- Permettent d'avoir un aperçu du type de trafic reçu (ou émis !) pour :
  - **Réagir en cas d'anomalie**
  - **Adapter et faire évoluer l'infrastructure** en cas de besoin

Exemple d'outil 1 : Snort

- IDS et IPS créé en 1998, édité par Sourcefire, aujourd'hui détenu par Cisco
- Diffusé sous licence GPL
- S'appuie sur des règles écrites par la communauté ou l'éditeur
- Peut détecter :
  - Des **scans de ports**
  - Des **dépassements de buffers**
  - Des tentatives d'**injections SQL**
  - Des **requêtes DNS** mal formées ou suspectes
  - Des **requêtes HTTP** mal formées ou suspectes
  - …

Snort par l'image



Services / Snort / Alerts

Snort Interfaces | Global Settings | Updates | Alerts | Blocked | Pass Lists | Suppress | IP Lists | SID Mgmt | Log Mgmt | Sync

Clear all interface log files

Alert Log View Settings

Interface to Inspect: WAN — Choose interface..

Auto-refresh view

1000 — Alert lines to display.

Save

Alert Log Actions: Download | Clear

Alert Log View Filter

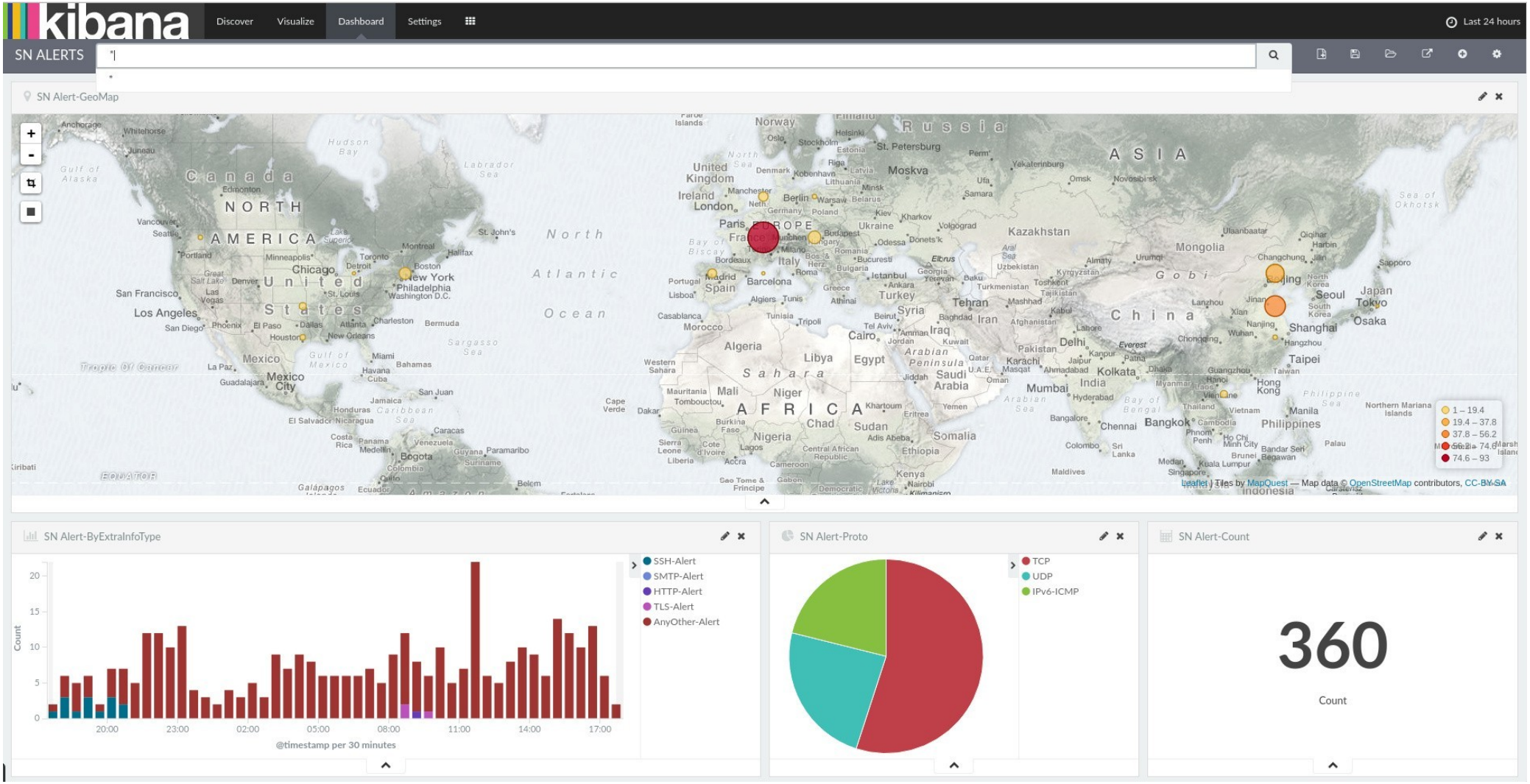
Last 1000 Alert Log Entries

Date	Pri	Proto	Class	Source IP	SPort	Destination IP	DPort	SID	Description
2017-07-23 20:49:52	1	UDP	A Network Trojan was Detected	66.240.205.34	1066		16464	1:31136	MALWARE-CNC Win.Trojan.ZeroAccess inbound connection
2017-07-22 06:15:49	2	UDP	Potentially Bad Traffic	163.172.17.76	54465		5060	140:26	(spp_sip) Method is unknown
2017-07-21 09:26:30	2	UDP	Potentially Bad Traffic	163.172.22.169	52428		5060	140:26	(spp_sip) Method is unknown
2017-07-21 01:03:28	2	UDP	Potentially Bad Traffic	163.172.17.76	46834		5060	140:26	(spp_sip) Method is unknown
2017-07-20 20:36:37	2	UDP	Potentially Bad Traffic	163.172.22.169	54788		5060	140:26	(spp_sip) Method is unknown
2017-07-20 08:31:30	2	UDP	Potentially Bad Traffic	163.172.17.76	59571		5060	140:26	(spp_sip) Method is unknown

Exemple d'outil 2 : SELKS

- Distribution tout-en-un comprenant basée sur :
  - **Suricata** (IDS/IPS multi-thread, concurrent et compatible avec Snort, apportant de nouvelles fonctionnalités comme la réputation d'IP)
  - Une stack **ELK** :
    - **Elasticsearch** : stockage, recherche et analyse des logs
    - **Logstash** : collecte et formatage des logs pour les injecter dans Elasticsearch
    - **Kibana** : création de requêtes et de dashboards
  - Une interface Web spécifique
- Édité par Stamus Networks et distribué sous licence GPLv3

SELKS par l'image



L’état des mises à jour du parc

- Pourquoi mettre à jour ?
  - Patcher des **failles de sécurité**
  - Patcher des **bugs**
  - Améliorer les performances

- La complexité de la gestion des mises à jour augmente en fonction :
  - Du **nombre** d’équipements/systèmes
  - De la **variété** des équipements/systèmes
  - Des **contraintes de production** (application ou système critique, plages de maintenances…)
  - De la **sévérité** des failles (CVE !)

Exemple d’outil 1 : à la main ?...

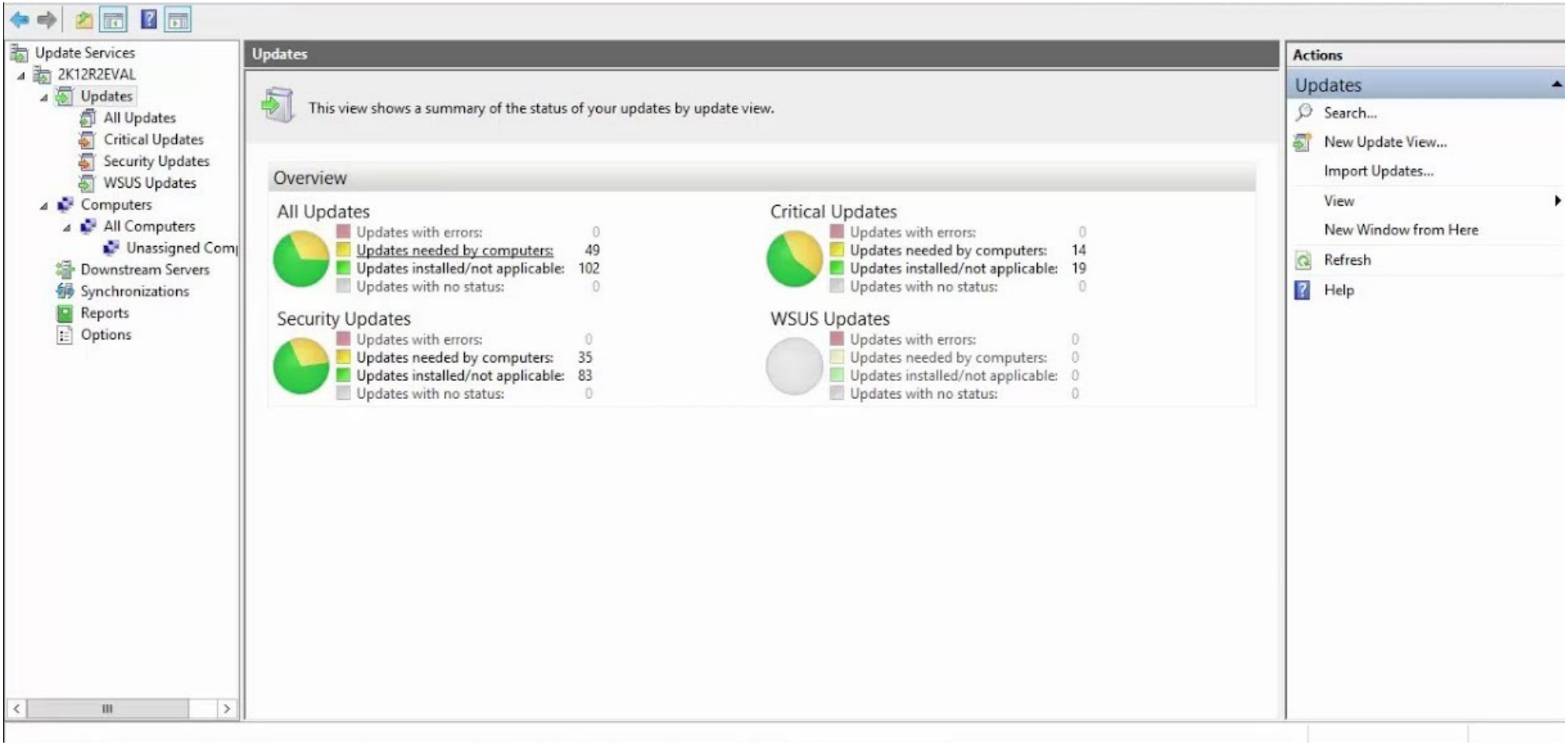
- Peu d’outils disponibles dans le monde Linux :(

- Il existe des plugins pour **Nagios/Centreon/etc** s’appuyant sur **apt/yum/pkg_add/zypper** pour vérifier la quantité de mises à jour disponibles
- **SUSE Manager** : outil de gestion d’infrastructure, permet le déploiement de machines et de configurations ainsi que le suivi des mises à jour du parc

Exemple d'outil 2 : WSUS

- **WSUS : Windows Server Update Services**
- Disponible dans **Windows Server**
- Permet la gestion des mises à jour du parc :
  - Récupère les mises à jour depuis Windows Update
  - Peut permettre la mise à jour de **machines isolées**
  - Affiche un **état du parc** (mises à jour en attente, mises à jour critiques en attente…)
  - Permet aux administrateurs de **filtrer** les mises à jour
  - Permet de **déployer** les mises à jour ou de les **planifier**

WSUS par l'image



Les failles de sécurité du parc

- Une faille de sécurité peut être due :
  - À une **implémentation logicielle** défectueuse
  - À une **mauvaise configuration**
  - À une négligence ou malveillance humaine

- Bien que souvent côté « Web » (mauvaise configuration Apache2 ou Nginx, version obsolète de PHP qui permet d'exploiter des bugs…), peuvent également se retrouver dans :
  - SSL (ex : Heartbleed)
  - RDP (Remote Desktop Protocol)
  - Des bases de données ouvertes en accès libre…

Exemple d'outil 1 : Nessus

- Outil créé en 1998, gratuit et Libre à l'époque
- Devenu **propriétaire** en 2005, édité par Tenable
- Permet de **scanner une infrastructure** pour détecter :
  - Des **vulnérabilités** (failles logicielles ou d'implémentation)
  - Des **mauvaises configurations** (SSH, bases de données, Web…)
  - Des **mots de passe par défaut** résiduels (!)
  - Des **URL** permettant d'accéder à des ressources qui ne devraient pas être accessibles/exister (ex : /admin, /test.php…)
- Génère des **rapports** dans différents formats (PDF, HTML) avec des recommandations
- Permet de définir des scans réguliers et des alertes

Nessus par l'image

SecurityCenter | Dashboard | Analysis | Scanning | Reporting | Assets | Workflow

Mobile | Add Component

Mobile Vulnerabilities

Plugin ID	Name	Severity	Total
65633	Apple iOS 6.1.3 Multiple Vulnerabilities	High	271
64287	Apple iOS 6.1 Multiple Vulnerabilities	High	192
62803	Apple iOS 6.0.1 Multiple Vulnerabilities	High	112
62242	Apple iOS 6.0 Multiple Vulnerabilities	High	70
60027	Apple iOS 5.1.1 Multiple Vulnerabilities	Critical	23
60028	Apple iOS 5.1 Multiple Vulnerabilities	High	20
60025	Apple iOS 5.0.1 Multiple Vulnerabilities	High	12
60026	Apple iOS 5.0 Multiple Vulnerabilities	High	12
62516	Windows Phone7 7.0.7392 Out-of-Date SSL Blacklist	Medium	1
62517	Windows Phone7 7.10.8107 Out-of-Date SSL Certificate Blacklist	Medium	1

Last Updated: 1 hour ago

Mobile Device Types

Last Updated: 1 minute ago

Vulnerable Users

User	Score	Info	Low	Medium	High	Critical	Total
[redacted]	56	2	0	0	12	2	16
[redacted]	37	4	0	0	9	1	14
[redacted]	37	3	0	0	9	1	13
[redacted]	34	3	0	0	8	1	12
[redacted]	34	2	0	0	8	1	11
[redacted]	31	4	0	0	7	1	12
[redacted]	31	1	0	0	7	1	9
[redacted]	31	1	0	0	7	1	9
[redacted]	31	1	0	0	7	1	9
[redacted]	31	3	0	0	7	1	11

Last Updated: 1 hour ago

Mobile Device Types

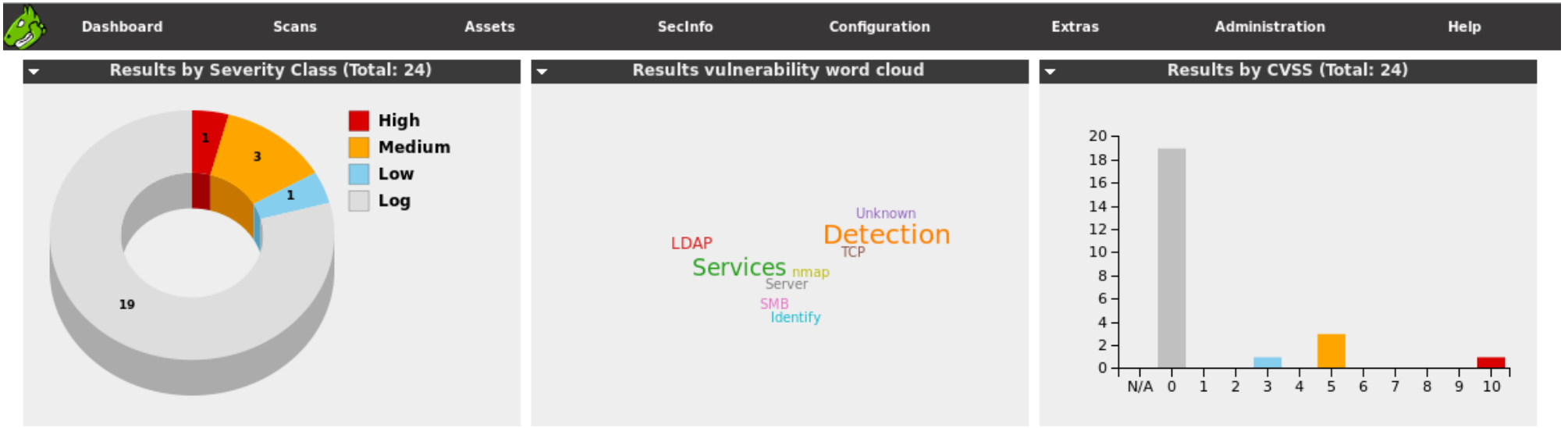
Model	Device Count	Total	High	Critical
WinPhone	11	13	0	0
iPhone	672	1077	389	16
iPad	417	724	300	7
BlackBerry	34	34	0	0
Android	72	72	0	0

Last Updated: 1 hour ago

Exemple d’outil 2 : OpenVAS

- **OpenVAS : Open Vulnerability Assessment System**
- Fork de Nessus, sous licence GPL
- Les plugins OpenVAS sont écrits en NASL (Nessus Attack Scripting Language)
- Dispose de dizaines de milliers de plugins
- Travaille de la même façon que Nessus :
  - Scan des infrastructures
  - Génération de rapports et de recommandations
  - Peut être automatisé et générer des alertes

OpenVAS par l'image



1 - 10 of 24

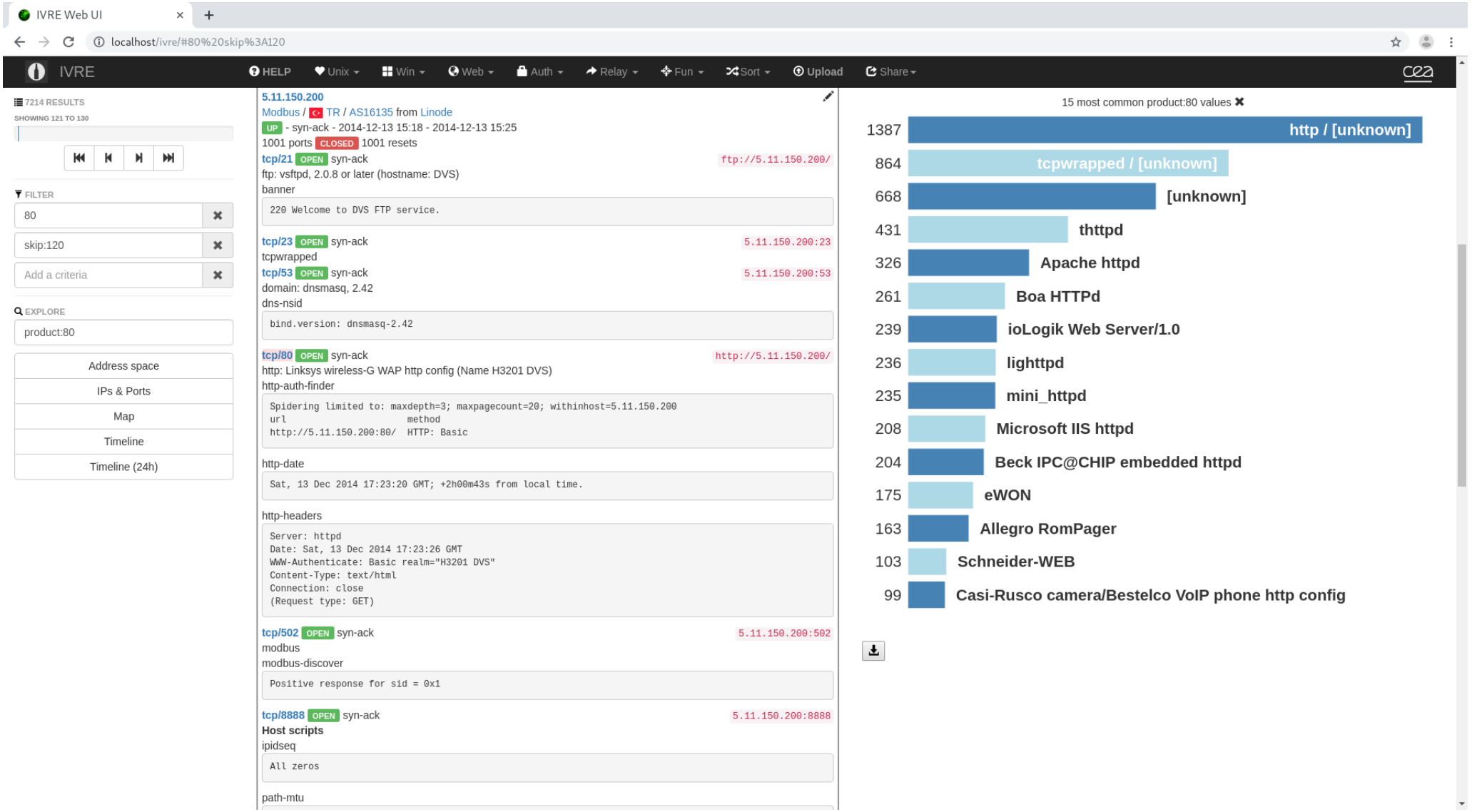
Vulnerability	Severity	QoD	Host	Location	Created
SMBv1 Unspecified Remote Code Execution (Shadow Brokers)	10.0 (High)	80%	172.16.161.138	445/tcp	Tue Apr 4 17:29:21 2017
Use LDAP search request to retrieve information from NT Directory Services	5.0 (Medium)	99%	172.16.161.138	3268/tcp	Tue Apr 4 17:29:25 2017
Use LDAP search request to retrieve information from NT Directory Services	5.0 (Medium)	99%	172.16.161.138	389/tcp	Tue Apr 4 17:29:27 2017
DCE Services Enumeration Reporting	5.0 (Medium)	80%	172.16.161.138	135/tcp	Tue Apr 4 17:29:31 2017
TCP timestamps	2.6 (Low)	80%	172.16.161.138	general/tcp	Tue Apr 4 17:29:24 2017
Traceroute	0.0 (Log)	80%	172.16.161.138	general/tcp	Tue Apr 4 17:27:19 2017
SMB/CIFS Server Detection	0.0 (Log)	80%	172.16.161.138	445/tcp	Tue Apr 4 17:27:19 2017
DCE Services Enumeration	0.0 (Log)	80%	172.16.161.138	135/tcp	Tue Apr 4 17:27:19 2017
SMB/CIFS Server Detection	0.0 (Log)	80%	172.16.161.138	139/tcp	Tue Apr 4 17:27:19 2017
DNS Server Detection (TCP)	0.0 (Log)	80%	172.16.161.138	53/tcp	Tue Apr 4 17:27:26 2017

Exemple d'outil 3 : IVRE

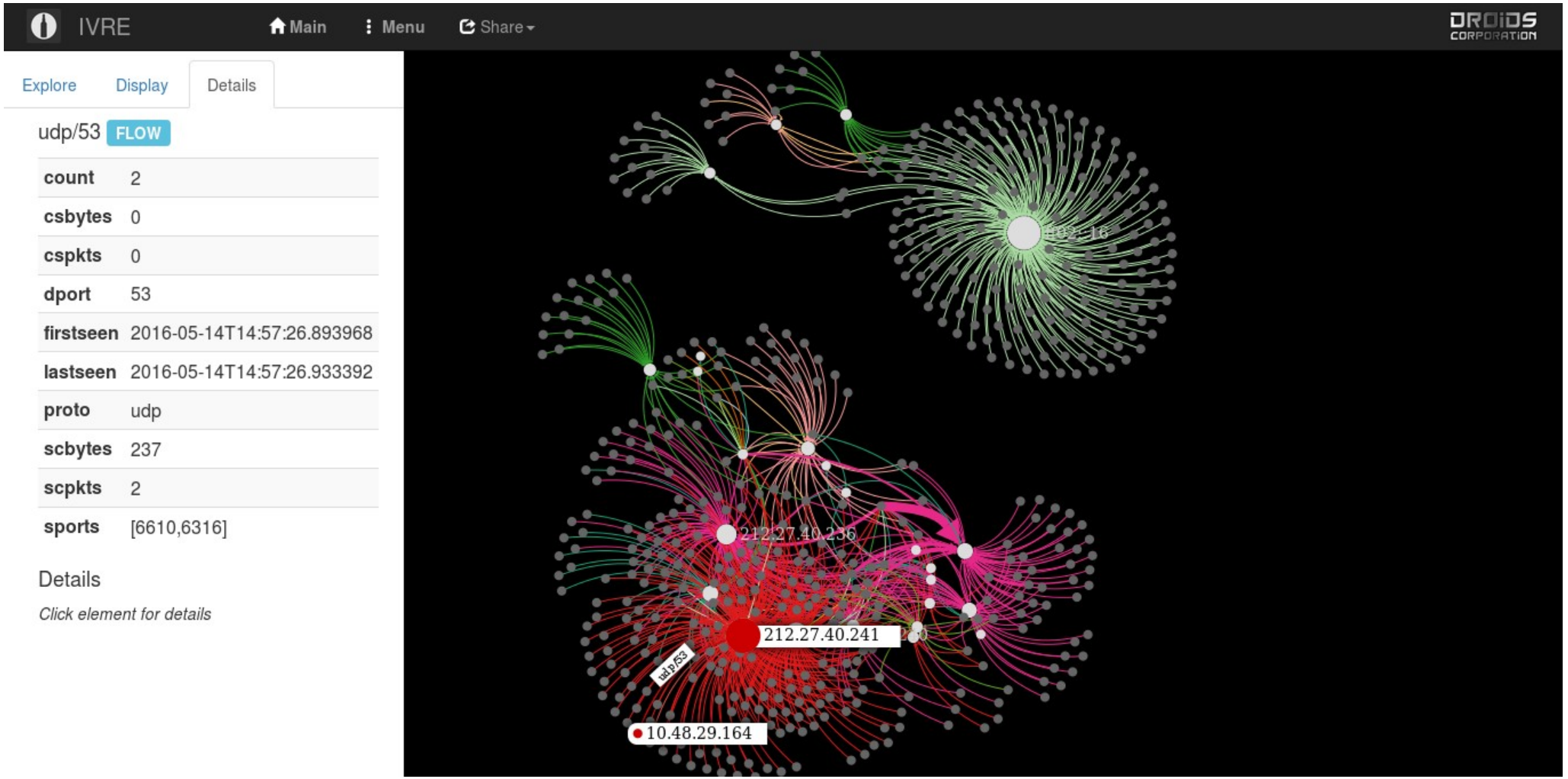
- **IVRE : Instrument de veille sur les réseaux extérieurs**
- *Aussi appelé DRUNK (Dynamic Recon of UNKnown networks)*
- Activement soutenu par le **CEA** (Commissariat à l'Energie Atomique)

- S'appuie sur des approches **passives** (NfDump, Argus…) et **actives** (Nmap, ZMap…)
- Permet d'analyser les flux, espaces d'adressage, ports ouverts et systèmes éventuellement mal configurés de son infrastructure
- Accessible en CLI, en Web et via une API Python

IVRE par l'image



IVRE par l'image (une 2e pour la route)

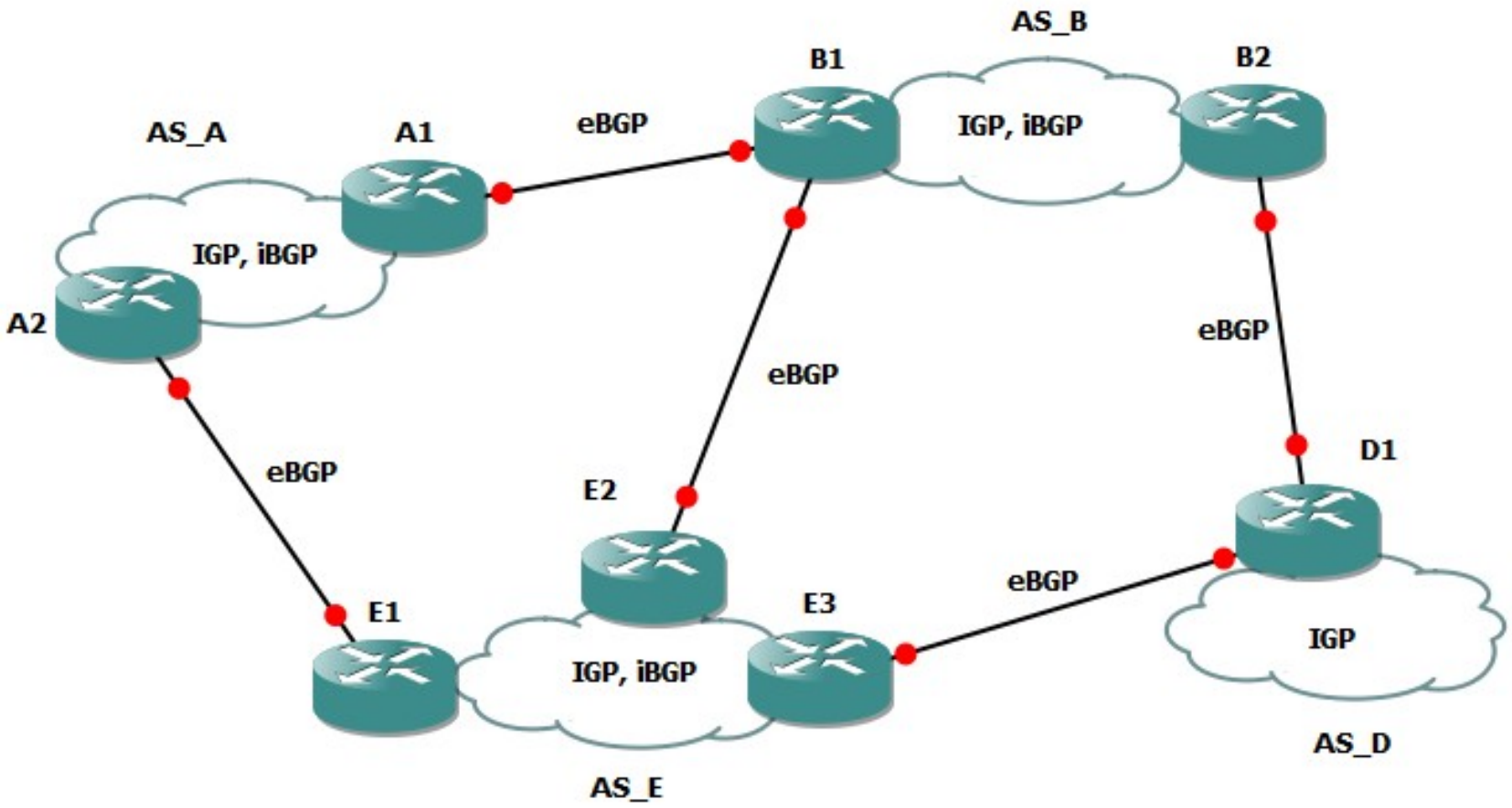


Annexe 3 : aux frontières du réseau

Rappels rapides sur BGP

- **EGP** (Exterior Gateway Protocol), protocole de routage dynamique
- Sert à interconnecter des **AS (Autonomous Systems)**
- Un AS peut être un **opérateur**, une **entreprise**, un **service public**
- Les AS montent des **sessions** entre eux et annoncent des **préfixes** (blocs d'IP publiques), soit en propre, soit au titre de **transitaire**
- **Protocole à vecteur de chemins**, BGP privilégie par défaut l'AS-path le plus court…
- … mais les règles de routage configurées par chaque AS sont avant tout **politiques** :
  - **Coût du trafic**
  - **Fonctionnalités** (anti-DDOS…)
  - **Affinités** avec telle entreprise, telle plateforme, tel état

Rappels rapides sur BGP

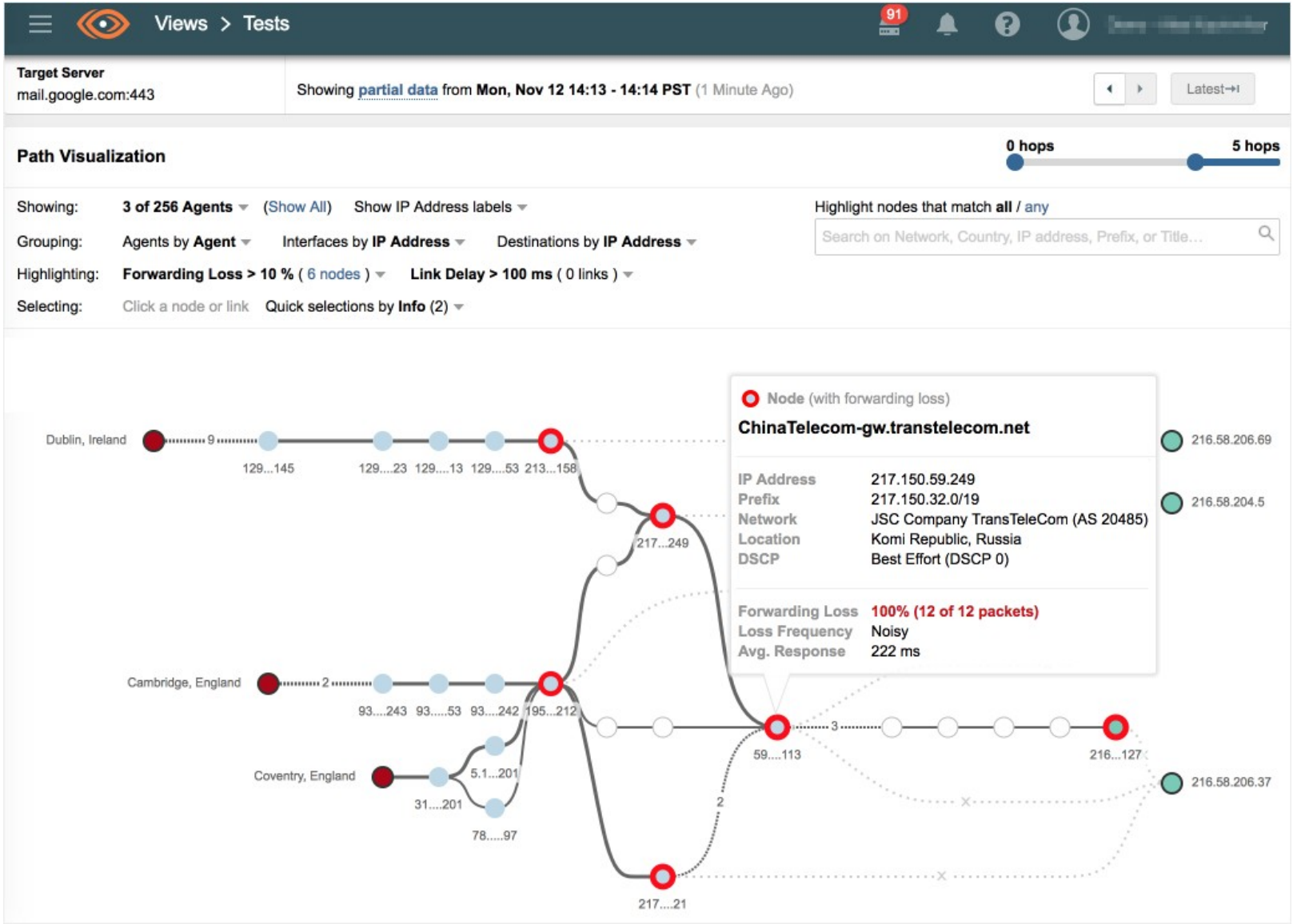


Pourquoi surveiller ses annonces BGP ?

- **Vos annonces BGP = la joignabilité de vos services !**
- Mauvaise configuration dans la chaîne ou **hijack** = trou noir ou instabilités

- Quelques examples (mais des dizaines par jour !) :
  - En 2008, Pakistan Telecom hijack Youtube au niveau mondial pendant 2h
  - En juin 2015, Telekom Malaysia annonce de nombreuses routes erronées, reprises par Level3… Le trafic mondial est impacté pendant une bonne partie de la journée
  - En avril 2017, l'AS 12389 (PJSC Rostelecom) annonce pendant quelques minutes de nombreux préfixes liés à des banques ou des organismes financiers…
  - En novembre 2018, Google est partiellement injoignable pendant 1h30

Un exemple de hijack BGP



Source : https://blog.thousandeyes.com/internet-vulnerability-takes-down-google/

Un autre exemple de hijack BGP

https://www.youtube.com/watch?v=IzLPKuAOe50

Exemple d'outil 1 : RIPE Atlas

- Réseau de sondes maintenu par le RIPE
- **Plusieurs milliers de sondes** dans le monde, hébergées gratuitement par des entreprises, des FAI, des particuliers…
- Les sondes fournissent des **métriques** agrégées par le RIPE (joignabilité, latence…) qui permettent de dresser une **cartographie de la qualité et de la performance du réseau IP**
- Chaque entité qui héberge une sonde obtient des crédits
- Chaque entité qui possède des crédits peut effectuer des **requêtes** pour effectuer des **tests** (ping, traceroute, DNS, NTP, HTTP…) depuis les sondes qui l'intéresse
- Disponible via de nombreux outils Web, une API Python, un client CLI…

RIPE Atlas par l'image

General Information | Probes | Map | LatencyMON | OpenIPMap Prototype | Results | Modification

Probe	ASN (IPv4)	ASN (IPv6)		Time (UTC)	RTT		Hops
2713	60706	60706		2016-11-18 10:52	33.192		14
2941	25394			2016-11-18 10:51	50.783		20
3055	6412			2016-11-18 10:53	150.683		15
3222	6829			2016-11-18 10:49	36.686		24
4166	50581			2016-11-18 10:52	39.533		16
4554	6703			2016-11-18 10:51	82.704		19
4952	3244			2016-11-18 10:51	35.700		19
6078	202040	202040		2016-11-18 10:47	9.279		14
6091	5459	5459		2016-11-18 10:50	9.719		14
6112	197216	197216		2016-11-18 10:52	33.767		11
6139	18106	18106		2016-11-18 10:47	216.946		19
10166	5379			2016-11-18 10:49	60.850		19
10282	49009	49009		2016-11-18 10:47	32.699		11
10312	11426			2016-11-18 10:49	116.443		29

Exemple d'outil 2 : BGPmon

- Entreprise fournissant un service de supervision des annonces BGP
- Propose les fonctionnalités suivantes :
  - Supervision en temps réel des **annonces de préfixes** avec plusieurs centaines de sondes dans le monde (permet de détecter des anomalies régionales)
    - Qui annonce vos préfixes ?
    - Est-ce que mes politiques de routages sont respectées ?
  - Supervision des **validations ROA** (Route Origin Authorization, permet de valider de manière cryptographique qu'un AS est bien autorisé à annoncer un préfixe)
  - Supervision des préfixes annoncés par votre AS

BGPmon par l'image

BGPMON

Welcome Andree Toonk | BGPMon API | Help | Configurations &

HOME | AUTONOMOUS SYSTEMS | PREFIXES | ALERTS | PEERMON

My Alerts

Alerts Details

Tools

On Friday April 22nd 2016 at 17:10 UTC we detected a Origin AS Change event for your prefix (199.16.156.0/23 *twitter*)
The detected prefix: 199.16.156.0/24, was announced by AS65021 (-Private Use AS-)

Alert description: Origin AS Change
Detected Prefix: 199.16.156.0/24
Detected Origin AS: 65021
Expected Origin AS: 13414

This alert was detected by 19 unique probes in 12 unique countries

- United States: 4 Peers
- Canada: 3 Peers
- Russian Federation: 2 Peers
- Korea, Republic of: 2 Peers
- Sweden: 1 Peers
- Bulgaria: 1 Peers
- Italy: 1 Peers
- Switzerland: 1 Peers
- Netherlands: 1 Peers
- Czech Republic: 1 Peers
- Germany: 1 Peers
- Slovenia: 1 Peers

UPDATE TIME (UTC)	UPDATE TYPE	PROBE ASN	PROBE LOCATION	PREFIX	AS PATH	CLEARED	DURATION ↓
2016-04-22 17:10:29	Update	AS24482	SE	199.16.156.0/24	24482 43531 200759 65021	2016-04-22 17:30:13	00:19:44
2016-04-22 17:10:16	Update	AS6881	CZ	199.16.156.0/24	6881 15685 6939 200759 65021	2016-04-22 17:20:29	00:10:13
2016-04-22 17:10:17	Update	AS49402	SI	199.16.156.0/24	49402 9119 6939 200759 65021	2016-04-22 17:20:30	00:10:13
2016-04-22 17:10:25	Update	AS395089	CA	199.16.156.0/24	395089 18451 6939 200759 65021	2016-04-22 17:20:25	00:10:00
2016-04-22 17:10:02	Update	AS63297	CA	199.16.156.0/24	63297 6939 200759 65021	2016-04-22 17:19:58	00:09:56
2016-04-22 17:10:15	Update	AS14866	CA	199.16.156.0/24	14866 6939 200759 65021	2016-04-22 17:19:58	00:09:43

Pilotage, gouvernance et hypervision

La gestion des alertes

SLA / GTI / GTR...

- **SLA : Service-Level Agreement**
  - **Engagement contractuel sur un niveau de service et de disponibilité**
  - Décrit les attendus du client, les temps de réponse aux incidents, le suivi des processus, les pénalités en cas de non-respect...
  - Exprimé en pourcentages (ex : 99,99 % = 8s / jour, ou 4min 22s / mois)

- **GTI : Garantie de temps d'intervention**
  - Délai contractuel dans lequel un incident doit avoir été **pris en charge**

- **GTR : Garantie de temps de rétablissement**
  - Délai contractuel dans lequel un incident doit avoir été **résolu**

SLA : une complexité croissante

- **La SLA d'un service dépend de celle du composant le plus faible**
  - Ex fictif :
    - SLA réseau = 99,999 % (26s / mois)
    - SLA stockage = 99,99 % (4min 22s / mois)
    - **SLA VMWare = 99,9 % (43min / mois)**

- **SLA ≠ perception client !**
  - Ex :
    - Un client loue 3 bases de données avec une SLA de 99,99 % / BDD
    - Si chaque BDD a une panne < 2 min dans le mois, SLA ok mais perception client = 3 pannes de son application dans le mois !

- **Le temps de détection et de prise en charge d'une panne est crucial !**

Différents types de maintenance

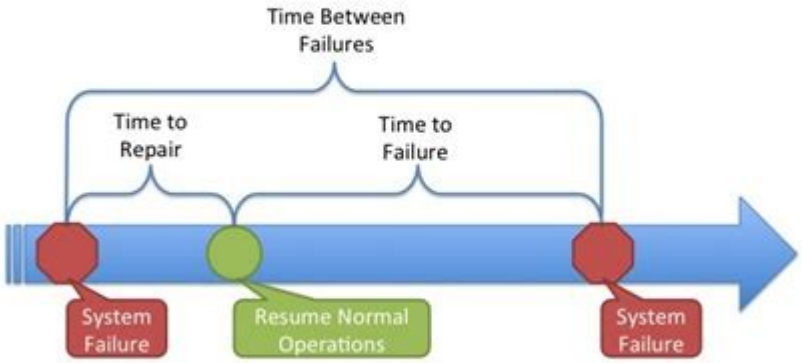
- **Maintenance curative :**
  - Consiste à intervenir sur un équipement ou un système après une panne, dans le but de réparer ou corriger.

- **Maintenance préventive :**
  - Consiste à prévoir des actions régulières sur un équipement ou un système afin d'assurer son entretien.
    - Ex : remplacer un disque dur tous les ans.

- **Maintenance prédictive :**
  - Basée sur des modèles d'analyse et des seuils, permet d'anticiper « juste à temps » les interventions nécessaires pour la réparation ou le remplacement d'un équipement ou d'un système.

MTBF, MTTR, MTTF, FIT...

- **MTTF : Mean Time To Failure**
- **MTTR : Mean Time To Repair**
- **MTBF : Mean Time Between Failure**
- **FIT : Failure In Time**
- **MDT : Mean Down Time**
- **...**



MFG	Model	Drive Size	Drive Count	Drive Days	Failures	Annualized Failure Rate
HGST	HMS5C4040ALE640	4TB	2,852	1,048,376	17	0.59%
HGST	HMS5C4040BLE640	4TB	12,746	4,674,986	57	0.45%
HGST	HUH728080ALE600	8TB	1,000	368,454	8	0.79%
HGST	HUH721212ALE600	12TB	1,560	327,080	5	0.56%
HGST	HUH721212ALN604	12TB	10,859	2,848,164	31	0.40%
Seagate	ST4000DM000	4TB	19,211	7,325,582	402	2.00%
Seagate	ST6000DX000	6TB	886	379,894	10	0.96%
Seagate	ST8000DM002	8TB	9,809	3,591,167	125	1.27%
Seagate	ST8000NM0055	8TB	14,447	5,242,891	225	1.57%
Seagate	ST10000NM0086	10TB	1,200	437,259	12	1.00%
Seagate	ST12000NM0007	12TB	37,004	12,721,076	1,156	3.32%
Seagate	ST12000NM0008	12TB	7,215	321,275	10	1.14%
Toshiba	MD04ABA400V	4TB	99	39,788	0	0.00%
Toshiba	MG07ACA14TA	14TB	3,619	564,829	10	0.65%
		Totals	**122,507**	**39,890,821**	**2,068**	**1.89%**

https://www.backblaze.com/blog/hard-drive-stats-for-2019/

Déterminer des seuils d'alerte

- **Déterminer le fonctionnement normal** du système (***ne pas se fier aveuglément aux données constructeur ou aux DAT fournis par les prestataires…***) :
  - Benchmarks, tests de résilience…
- **Définir des seuils** en accord avec :
  - Les résultats des tests et les données constructeurs
  - Les engagements contractuels avec les clients
  - **Les capacités d'interventions des équipes !** (temps d'intervention, outils à disposition…)
- **Adapter** les seuils d'alerte **au cours de la vie du système :**
  - Evolutions des infrastructures, nouveaux clients, nouvelles SLA…

Exemples de seuils d'alerte

- **Taux de remplissage d'un espace disque :**
  - Tenir compte de l'espace total, du type et de la volatilité des données, des besoins applicatifs et/ou utilisateurs (traitements ponctuels lourds, fichiers temporaires importants…) et de la vitesse de remplissage (nécessite d'avoir des métriques pour adapter dans le temps)
  - Exemple classique : Warning à 80 %, Critical à 90 %

- **Accessibilité d'un site web :**
  - Sur un infrastructure composée de 4 BDD et 20 Frontaux :
    - **Choix 1** : Alerte pour chaque serveur qui tombe
    - **Choix 2** : Alerte uniquement quand le site devient inaccessible
    - **Choix 3** : Mix des deux
      - On détermine à partir de quand le risque de panne totale devient trop important
      - Ou on détermine à partir de quand l'infrastructure ne tient plus la charge

La gestion des alertes

- Supposons une infrastructure composée d'un serveur de supervision, d'une base de données et d'un poller :
  - ***Que se passe-t-il si le poller devient injoignable ?***
    - **Plus aucun check, plus aucune alerte !**
  - → **Il est impératif d'avoir une infrastructure distribuée, où les différents composants veillent les uns sur les autres.**

- Supposons une infrastructure distribuée, composée de plusieurs pollers et bases de données réparti·e·s sur 2 datacenters :
  - ***Que se passe-t-il en cas de DDOS ou de défaillance du/des transitaires ?***
    - **La supervision peut ne rien détecter, ou ne pas pouvoir alerter !**
  - → **Il est vivement recommandé de mettre en place une seconde supervision externe (un VPS chez un hébergeur peut suffire).**

Envoyer des alertes

- **Une alerte non envoyée = un incident non-traité, voire une situation qui dégénère**

- Il est impératif de prévoir **plusieurs moyens d'envoi** et une bascule automatique en cas d'indisponibilité de l'un deux :
  - SMS via une clé 4G ou un service Web
  - Mail, messagerie instantané…
  - Pager (!)

- Dans un datacenter, la couverture mobile est très mauvaise…
- … mais en cas d'incident réseau, les mails d'alertes ne passeront pas !

- **Ne pas centraliser l'envoi des alertes sur un seul composant !**

La gestion des escalades

- Si une alerte ne peut être traitée par le premier niveau de prise en charge, elle doit être « escaladée ».

- Dans beaucoup d'entreprises (type SSII), on parle souvent de :
  - **N1 : gestion des alertes et réponse aux incidents sur procédure simple**
  - **N2 : escalade technique, personnes qualifiées et diagnostique approfondi**
  - **N3 : expertise technique (parfois R&D ou direction technique)**

- L'escalade d'une alerte n'est pertinente que si :
  - Elle est **justifiée** (risque d'engorgement des niveaux supérieurs)
  - Elle est dirigée vers un **service compétent** (nécessite une gestion des compétences et une organisation adaptées)

La gestion des astreintes

- En dehors des heures ouvrées / horaires de service, la prise en charge des alertes reste **impérative**.

- Il est possible de définir contractuellement :
  - Les **périmètres couverts** par l'astreinte
  - Les **délais d'interventions** si différents
- En astreinte, la gestion des **escalades** reste primordiale :
  - **Escalades techniques** (peuvent nécessiter plusieurs personnes d'astreinte)
  - **Escalades managériales/opérationnelles** (en cas de prise de décisions majeures)

- Une entreprise installée sur plusieurs continents peut ne pas avoir besoin d'astreinte !

Incidents VS problèmes

- **Incident : évènement ponctuel, avec ou sans impact**
- **Problème : incidents répétés, avec ou sans impact, sur les mêmes composants**

- Incidents et problèmes ne sont **pas uniquement liés à l'infrastructure** : astreinte non prise en charge, procédure erronée..

- Certains outils permettent de remonter les problèmes, mais des **revues régulières** par les responsables de services sont impératives
- **Un incident doit être traité, un problème doit donner lieu à un plan d'action**

Gouvernance

Définition

*« La gouvernance des technologies de l'information a pour objectif de **mettre en œuvre la stratégie de l'entreprise** et notamment de saisir les différentes opportunités de développement qui s'offrent à l'entreprise. Ainsi **la gouvernance des TI participe à la création de valeur, permet d'optimiser l'utilisation des ressources informatiques et de gérer les risques liés à sa mise en œuvre.** »*

Source : Wikipédia

KPI : définition

- **KPI : Key Performance Indicator**

- **Indicateur mesurable d'aide décisionnelle,** défini **en phase avec la stratégie** de l'entreprise

- Peut être :
  - **Quantitatif** (de façon objective, ex : nombre de tickets résolus par mois)
  - **Qualitatif** (de façon subjective, ex : « bon », « améliorable »…)

- Un succès peut être **l'aboutissement d'un projet** ou le **maintien en conditions opérationnelles** d'un service (incidents traités, tickets suivis…)

KPI : quelques exemples de métriques

- **Disponibilité :**
  - Temps de résolution des incidents avec impact
  - Quantité de maintenances sur un composant
- **Performances :**
  - Nombre de requêtes par seconde sur une BDD
  - Délais de livraison de nouveaux serveurs
- **Pilotage :**
  - Quantité de tickets ouverts par client
  - Temps de prise en charge et de traitement
- **Sécurité :**
  - Tentatives d'intrusion
  - Failles découvertes et délais avant correction

Quels outils pour suivre ses KPI ?

- Deux grandes catégories :
  - Les outils d'**extraction**
    - Par **« batchs »** réguliers, sur de grosses volumétries, pouvant effectuer des conversions, des transformations…
    - En **« temps-réel »**, en scrutant les échanges inter-applications pour suivre des indicateurs (ex : croiser le stock de serveur disponibles (application de gestion du stock) et le nombre de commandes en cours (« pipe » commercial) pour déterminer la marge disponible et les commandes à prévoir)
  - Les outils de **présentation**
    - **Tableurs**
    - **Tableaux de bord** (spécifiques au métier ou génériques)

Capacity Planning : définition

*« L'objectif de la gestion de la capacité est de garantir que l'infrastructure informatique est fournie **au bon moment**, **au bon prix** et **en quantité adéquate** pour tenir la qualité de service en alignement avec **les besoins métiers**. »*

Source : Wikipédia

Gérer un CP : au carrefour des métiers

- La gestion de capacité repose sur plusieurs activités :
  - La collecte de **mesures**
  - La production d'**indicateurs** et la définition de **seuils**
  - Le **suivi des demandes** (pour le **« run »** comme pour le **« build »**)
  - La **planification**, avec les équipes techniques et la gouvernance

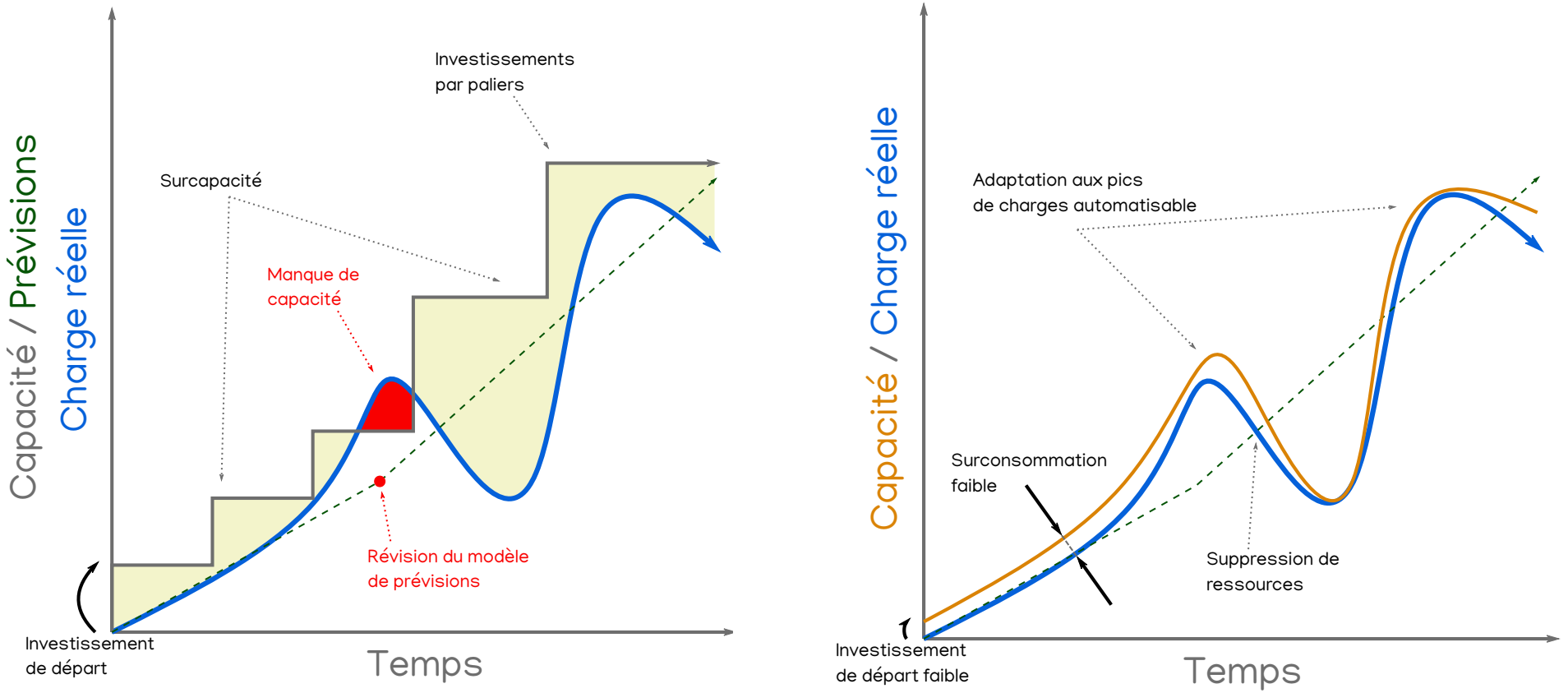
- Elle doit tenir compte :
  - Des **ressources matérielles** (serveurs, licences…)
  - Des **projets** en cours et à venir
  - Des **enveloppes budgétaires** actuelles et à venir
  - Des **équipes** et des **compétences**

Gérer un CP : au carrefour des équipes

- Un Capacity Planning doit être géré **conjointement par les équipes techniques et la gouvernance** (responsables clients, chefs de projets et direction)
  - Les équipes techniques connaissent les plateformes mais n'ont pas forcément connaissance de la **stratégie** de l'entreprise ou de l'arrivée de nouveaux **projets clients**
  - La gouvernance n'a pas forcément conscience des enjeux de tel ou tel investissement sur la **qualité** ou le **taux de disponibilité** du service

- Un investissement (financier, matériel, humain…) :
  - Trop faible = dégradation des conditions de travail, de la disponibilité et des performances du service, des **capacités de vente**, de la **satisfaction client**…
  - Trop important = une diminution de la **marge** de l'entreprise, des ressources mal exploitées…

Gérer un CP : CAPEX VS OPEX



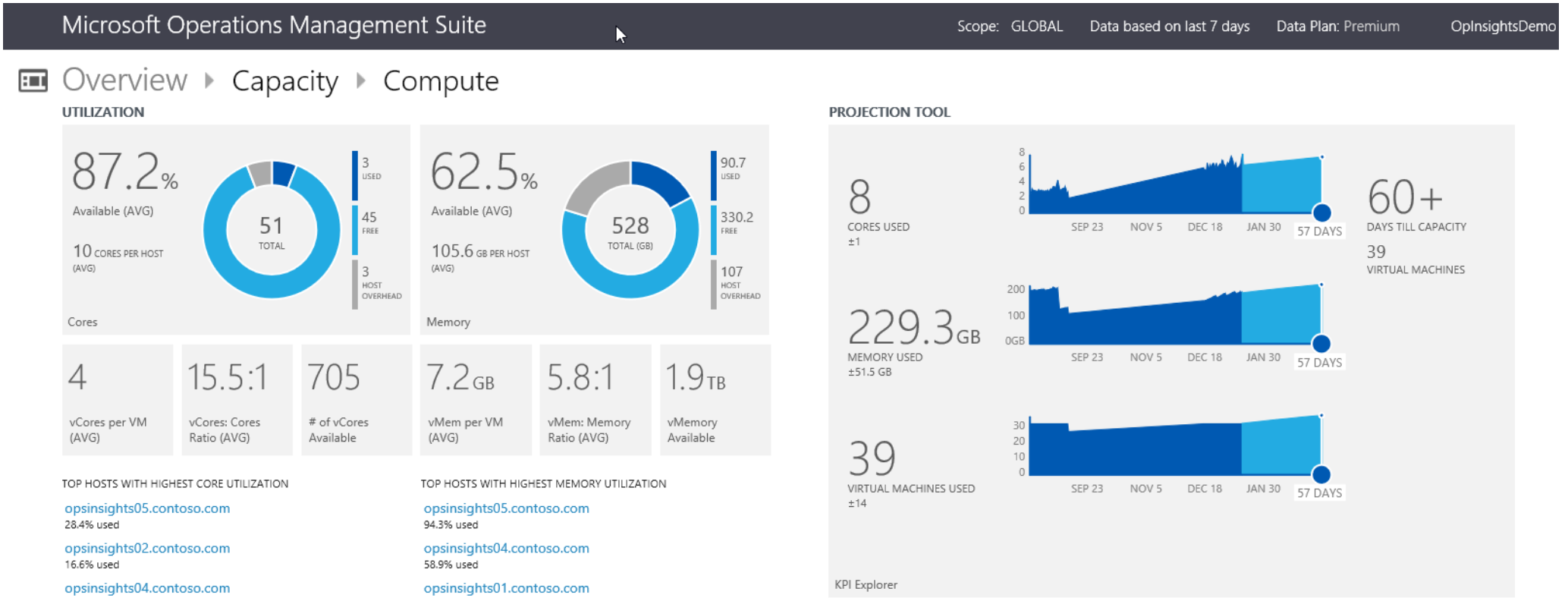
Source : Module de Cloud Computing, par OVH (Crédit : Pierre Gaxatte)

Quels outils pour gérer son CP ?

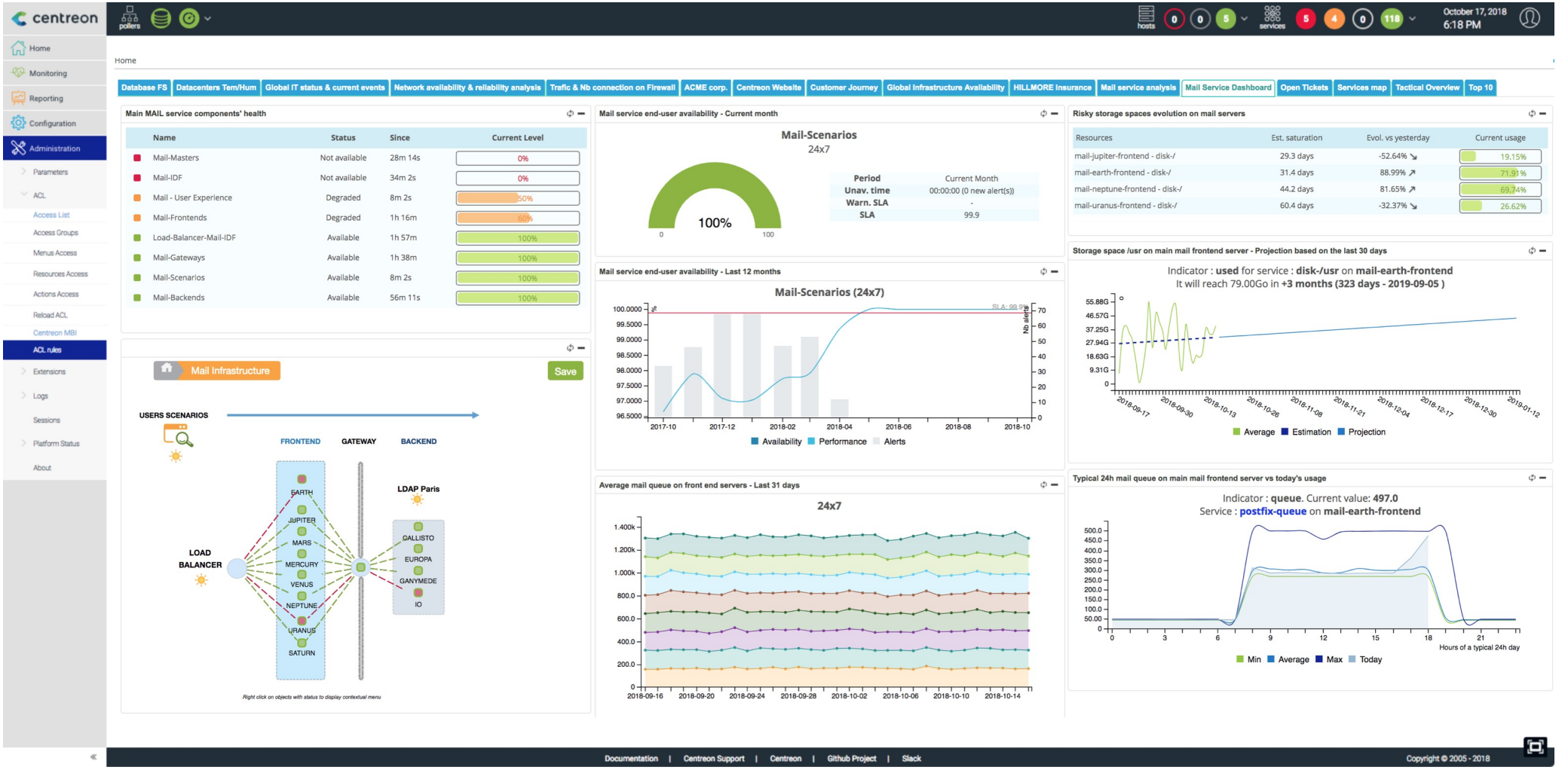
- D'un point de vue **« pilotage »** :
  - La définition et la communication de la **stratégie** de l'entreprise
  - Des revue régulières des **projets** et des **commandes** clientes
  - Un suivi des **compétences** des équipes

- D'un point de vue **« technique »** :
  - Des revues régulières des outils de **supervision** et de **métrologie**
  - Certains outils de supervision ou de pilotage d'infrastructures permettent de visualiser les tendances d'évolution des infrastructures, et d'anticiper les modifications requises (ex : rajouter une baie de disques dans 2 mois)
  - La création de **dashboards** métiers spécifiques (en s'appuyant par exemple sur Kibana ou Grafana)
  - Une **veille technologique** avancée : nouveaux outils, matériels…

Quelques exemples d'outils de CP



Quelques exemples d'outils de CP



Hypervision

Définition

*« En informatique, l'hypervision est la **centralisation des outils de supervision** d'infrastructure, d'applications et **de référentiels** (par exemple, la base de données configuration management database - CMDB). Il s'agit d'un outil **d'agrégation** ou de console unique. »*

Source : Wikipédia

Principe de l’hypervision

- Un SI complexe devant fournir plusieurs services et composer avec plusieurs équipes et applicatifs/outils = plusieurs outils de supervision, de ticketing…

- **Trop d’interfaces = pertes de temps et d’informations**

- Une console d’hypervision permet généralement de retrouver :
  - Un **bac à évènement** complet (alertes de supervision, logs, tickets en cours…)
  - Une **corrélation des évènements**
  - Des **indicateurs** (statistiques, métriques importantes) permettant d’avoir une **vision globale** de tout le SI

Les avantages de l'hypervision

- Une **console centralisée** avec des **vues adaptées** à chaque besoin (gouvernance, technique, client…)
- Des **indicateurs** déjà calculés et exploitables pour les KPI
- Des **corrélations d'évènements** permettant d'identifier des problèmes complexes

- Exemple de corrélation possible :
  1) Une forte charge réseau In/Out est détectée sur les plateformes (supervision)
  2) Une attaque DDOS est identifiée (outils externes + alertes prestataires)
  3) Des systèmes de fichiers sont soumis à une forte charge en lecture à une heure inhabituelle (supervision)

     ➔ ***Tentative d'extraction de données masquée dans le flux ?***

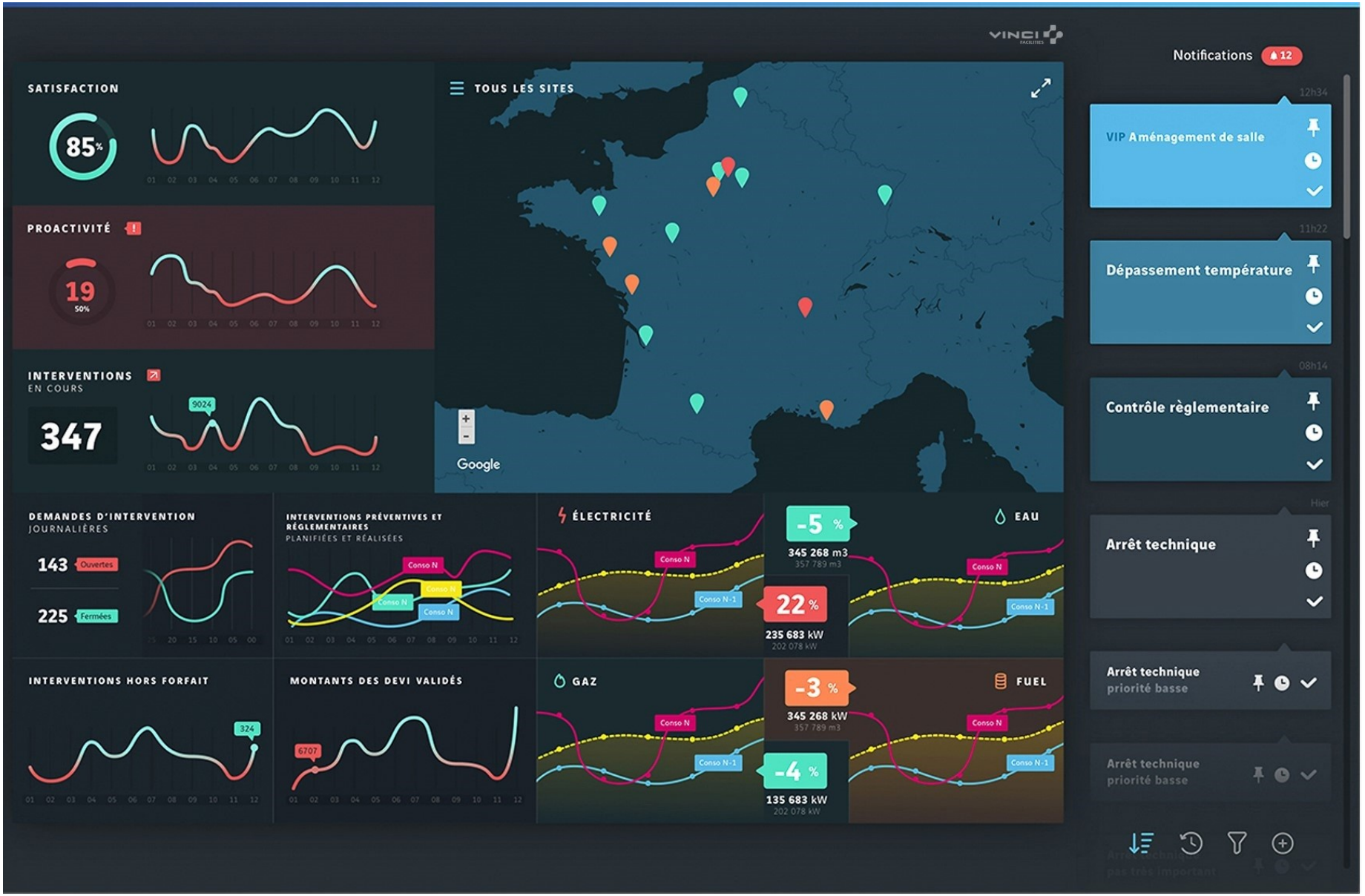
Les inconvénients de l'hypervision

- Un domaine assez peu connu
- Peu d'outils disponibles
- Une **complexité** de mise en place importante et croissante en fonction de la quantité d'outils à interfacer
- **Connecter des outils, c'est bien, mais réussir à les exploiter, c'est mieux !**
  - Identifier les informations importantes à mettre en avant est difficile
  - Créer des règles de corrélation d'évènements est complexe, car l'infrastructure évolue sans cesse

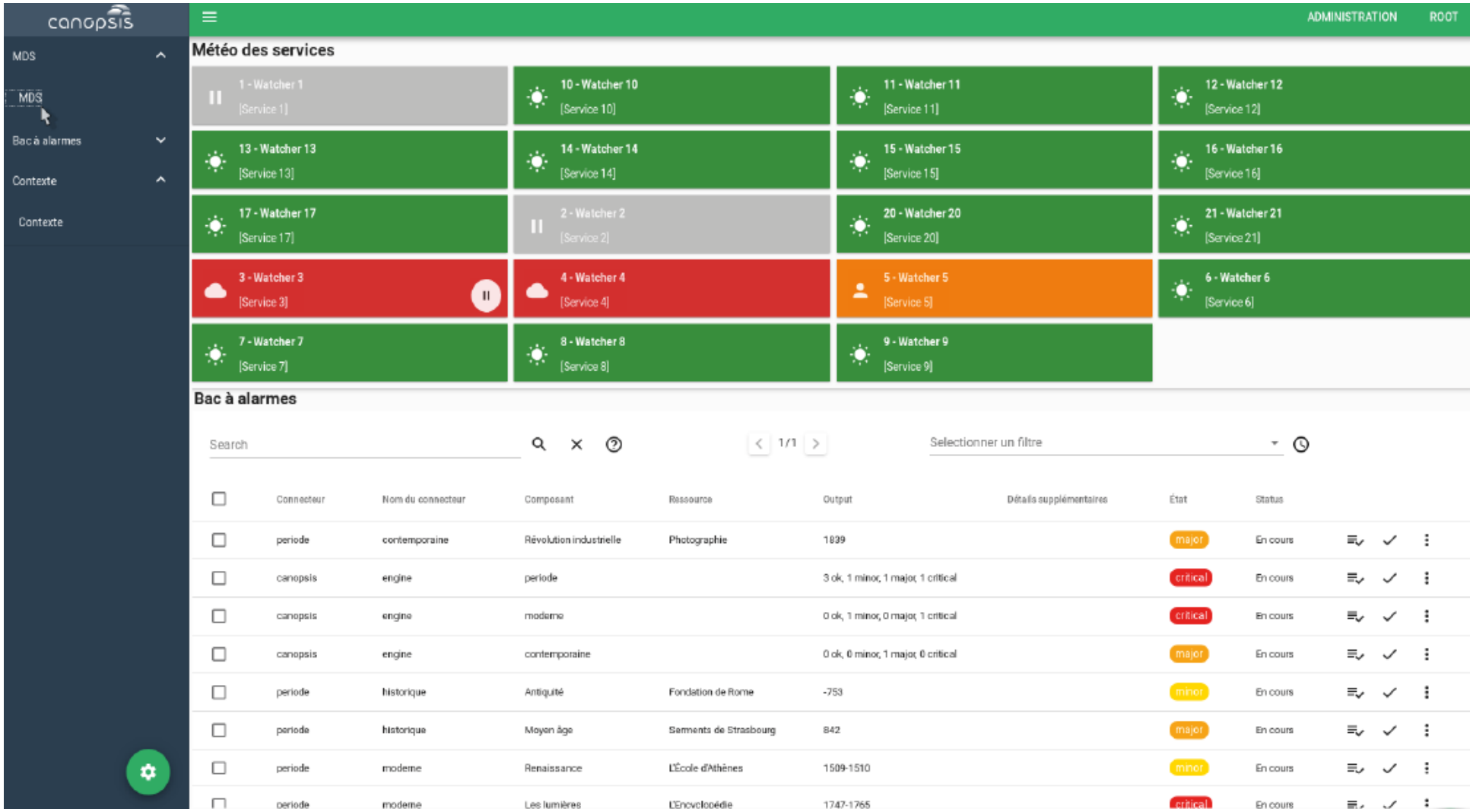
Les outils d’hypervision

- Aujourd’hui, plusieurs éditeurs de solutions de supervision commencent à intégrer des outils d’hypervision : Centreon, Zabbix…
- Des éditeurs proposent des solutions clé-en-main : Canopsis (de Capensis)
- D’autres proposent des solutions sur-mesure intégrées au SI (plus complexes, plus coûteuses mais potentiellement plus efficaces) : BMC, IBM, HP…
- Pour certains besoins ou secteurs industriels, on peut également trouver des outils et consoles spécifiques

Un exemple de console d'hypervision spécifique



Un autre exemple de console d'hypervision



Conclusion

In the end…

- Supervision ?
  - **Vérification de l'état d'un composant ou d'un service**

- Métrologie ?
  - **Mesure des performances ou des capacités dans le temps**

- Mettre en place des solutions de supervision :
  - Établir un **cahier des charges** avec toutes les parties prenantes (gouvernance **ET** opérations)
  - Procéder à une **étude technique comparative**
  - Choisir l'**outil adapté** et y mettre les **moyens** (humains et financiers)

Contact

Florian Haller-Casagrande

Mail : florian@haller-casagrande.fr

LinkedIn : https://www.linkedin.com/in/florianhc