R303 : Services Réseaux Avancés / Supervision de Réseau

Guillaume Urvoy-Keller

September 8, 2022

Objectif

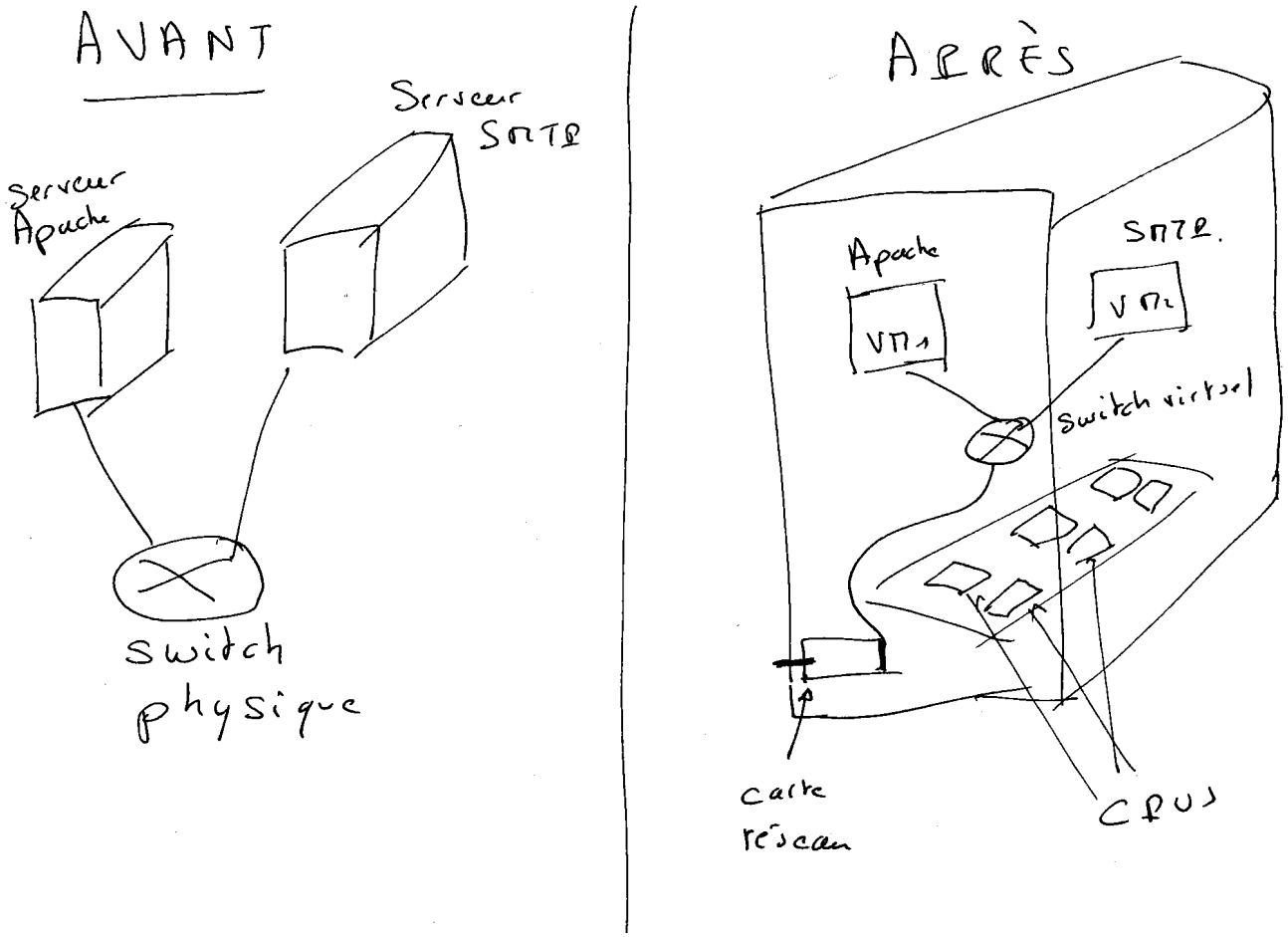
Supervision

- Réseau: Wireshark, ping, traceroute, smokeping
- Système/Application : SNMP, collectd, netdata, syslog

Automatisation

- Ansible

Supervision : la frontière floue réseau ou système



Organisation du cours

CM/Cours

- Intro
- Systemd et Apt
- Supervision réseau : ping, traceroute
- Supervision système : SNMP
- Automatisation tâches administration : Ansible

TD/TP

- TDs
  - Smokeping : analyse de mesures actives
  - SNMP et Ansible
  -
- TPs
  - Wireshark
  - Supervision de Réseau SNMP, netdata
  - TP Ansible

Organisation du cours

- Notation : 30 % TP et 70 % examen final
- Examen final comporte :
  - des questions de cours
  - **des questions sur les TPs** $\Rightarrow$ vous devez faire un compte-rendu électronique pour vous (+ pour l'encadrant suivant ses demandes)

M3105 - Administration des Serveurs Linux: Systemd, Paquets

Guillaume Urvoy-Keller

September 8, 2022

Systemd (System Daemon)

"Rappel" sur les services

- Appelés aussi démon, d'où leur nom, souvent avec un "d" : sshd, snmpd
- Un service est démarré via un script
  - /etc/init.d/apache2 start – OBSOLETE
  - service apache2 start – OBSOLETE
  - systemctl start apache2
- Les services écrivent leurs messages dans le log général /var/log/syslog → **c'est là qu'il faut aller voir pour debugger**

```
root@buster:/home/vagrant# grep apache2 /var/log/syslog
Sep 21 18:58:00 buster systemd[1]: apache2.service: Control
     process exited, code=exited, status=1/FAILURE
Sep 21 18:58:00 buster systemd[1]: apache2.service: Failed
    with result 'exit-code'.
Sep 21 19:35:37 buster systemd[1]: apache2.service:
    Succeeded.
Sep 21 19:35:43 buster systemd[1]: apache2.service: Unit
    cannot be reloaded because it is inactive.
```

Systemd: la gestion des services

- Adopté par toutes les distributions Linux depuis 2015: RHEL 7, CentOS sans oublier Debian et Ubuntu.
- Remplace SysVInit, basé sur des scripts dans /etc/init.d/...
- Fait beaucoup plus :
  - la gestion des ressources,
  - l'arrêt et le démarrage des services,
  - la détection des périphériques,
  - la journalisation,
  - la virtualisation par conteneur
- Projet initié par Lennart Poettering, un ingénieur de Red Hat.

Systemd

- Le premier processus démarré (PID=1)
- ..qui active ensuite les autres
- Les scripts dans SysVInit pouvaient être lents si blocage car exécution séquentielle :
  Ex: attente de l'activation de la carte réseau
- Dans Systemd introduit la notion d'unité
  - Peut être vue comme un container bas niveau dans lequel s'exécute ... un service
  - ou autre chose, par exemple une socket, le montage d'un système de fichier
- Unités peuvent être parallélisées $\rightarrow$ rapidité du boot!!!
- Gestion des blocages
- Journalisation
- Gestion VM et containers via libvirt et cgroups

Systemd: les fichiers clefs

On liste ce qui a été installé dans le paquet systemd.

Ce qui va nous intéresser : systemctl (= system control) et journalctl

```
root@buster:/home/vagrant# dpkg -L systemd
/bin
/bin/journalctl
/bin/loginctl
/bin/networkctl
/bin/systemctl
/bin/systemd-ask-password
/bin/systemd-escape
/bin/systemd-inhibit
/bin/systemd-machine-id-setup
/bin/systemd-notify
```

Systemd: les fichiers clefs

Les fichiers des unités dans /usr/lib/systemd/system.

```
root@buster:/home/vagrant# ls /usr/lib/systemd/system/
[...]
network-pre.target  systemd-halt.service
[...]
killprocs.service       sudo.service
[...]
ssh.service     syslog.socket
```

Extensions correspondent au type de l'unité : .service, .socket, .mount, .path....

Un exemple d'unité : le script pour Apache2

Le langage est déclaratif (décrit ce qui doit être fait) et non impératif (comment le faire) → plus compact (peu de code)

```
root@buster:/home/vagrant# more  /usr/lib/systemd/system/apache2.service
[Unit]
Description=The Apache HTTP Server
After=network.target remote-fs.target nss-lookup.target
Documentation=https://httpd.apache.org/docs/2.4/

[Service]
Type=forking
Environment=APACHE_STARTED_BY_SYSTEMD=true
ExecStart=/usr/sbin/apachectl start
ExecStop=/usr/sbin/apachectl stop
ExecReload=/usr/sbin/apachectl graceful
PrivateTmp=true
Restart=on-abort

[Install]
WantedBy=multi-user.target
```

Un exemple d'unité : le script pour Apache2

- Unit : Description et règles de dépendance vis à vis des autres unités
  Ex: démarrer le serveur après que le réseau soit opérationnel
- Service : commande effectivement lancée au démarrage via un

```
root@buster:/home/vagrant# systemctl start apache2
root@buster:/home/vagrant# systemctl stop apache2
root@buster:/home/vagrant# systemctl reload apache2
```

- WantedBy indique que le service est nécessaire à la cible multi-user.target = niveau d'exécution sans interface graphique.

Un exemple d'unité : le script pour Apache2

```
root@buster:/home/vagrant# systemctl status apache2
 apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
      enabled)
   Active: active (running) since Mon 2020-09-21 19:08:01 GMT; 6min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 2817 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 2821 (apache2)
    Tasks: 55 (limit: 545)
   Memory: 5.1M
   CGroup: /system.slice/apache2.service
           I--2821 /usr/sbin/apache2 -k start
           I--2822 /usr/sbin/apache2 -k start
           I-- 2823 /usr/sbin/apache2 -k start

Sep 21 19:08:01 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 19:08:01 buster systemd[1]: Started The Apache HTTP Server.
```

Des infos sur l'état du serveur, mais aussi : numéro du processus, fichier de log qui le concerne, et si il démarre au boot (=enabled)

Activation /désactivation au boot

```
root@buster:/home/vagrant# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/
     systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service -> /lib/
     systemd/system/apache2.service.
root@buster:/home/vagrant# systemctl disable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/
     systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable apache2
Removed /etc/systemd/system/multi-user.target.wants/apache2.service.
```

Afficher tous les services

```
root@buster:/home/vagrant# systemctl -t service
UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
apache2.service                     loaded active running The Apache HTTP Server
console-setup.service               loaded active exited  Set console font and keymap
cron.service                        loaded active running Regular background program
     processing daemon
getty@tty1.service                  loaded active running Getty on tty1
ifup@eth0.service                   loaded active exited  ifup for eth0
ifupdown-pre.service                loaded active exited  Helper to synchronize boot
     up for ifupdown
keyboard-setup.service              loaded active exited  Set the console keyboard
     layout
kmod-static-nodes.service           loaded active exited  Create list of required
     static device nodes for the current kernel
networking.service                  loaded active exited  Raise network interfaces
rsyslog.service                     loaded active running System Logging Service
ssh.service                         loaded active running OpenBSD Secure Shell server
[...]
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
```

Journalisation

Systemd démarre un service de journalisation qui capture tous les messages par unité, mais aussi le log du boot et le log global (syslog)

```
root@buster:/home/vagrant# journalctl -u apache2
-- Logs begin at Mon 2020-09-21 15:06:35 GMT, end at Mon 2020-09-21 19:21:48 GMT. --
Sep 21 18:58:00 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 18:58:00 buster apachectl[2534]: (98)Address already in use: AH00072:
     make_sock: could not bind to address [::]:80
Sep 21 18:58:00 buster apachectl[2534]: (98)Address already in use: AH00072:
     make_sock: could not bind to address 0.0.0.0:80
Sep 21 18:58:00 buster apachectl[2534]: no listening sockets available, shutting down
Sep 21 18:58:00 buster apachectl[2534]: AH00015: Unable to open logs
Sep 21 18:58:00 buster apachectl[2534]: Action 'start' failed.
Sep 21 18:58:00 buster apachectl[2534]: The Apache error log may have more
     information.
Sep 21 18:58:00 buster systemd[1]: apache2.service: Control process exited, code=
     exited, status=1/FAILURE
Sep 21 18:58:00 buster systemd[1]: apache2.service: Failed with result 'exit-code'.
Sep 21 18:58:00 buster systemd[1]: Failed to start The Apache HTTP Server.
Sep 21 19:08:01 buster systemd[1]: Starting The Apache HTTP Server...
Sep 21 19:08:01 buster systemd[1]: Started The Apache HTTP Server.
```

Journalisation - le boot

```
root@buster:/home/vagrant# journalctl -b
-- Logs begin at Mon 2020-09-21 15:06:35 GMT, end at Mon 2020-09-21 19:35:56 GMT. --
Sep 21 15:06:35 buster kernel: Linux version 4.19.0-5-amd64 (debian-kernel@lists.
     debian.org) (gcc version 8.3.0 (Debian 8.3.0-7)) #1 SMP Debian 4.19.37-5
     (2019-06-19)
Sep 21 15:06:35 buster kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-4.19.0-5-amd64
     root=UUID=b9ffc3d1-86b2-4a2c-a8be-f2b2f4aa4cb5 ro net.ifnames=0 quiet
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating
     point registers'
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE
     registers'
Sep 21 15:06:35 buster kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX
     registers'
Sep 21 15:06:35 buster kernel: x86/fpu: xstate_offset[2]:  576, xstate_sizes[2]:  256
Sep 21 15:06:35 buster kernel: x86/fpu: Enabled xstate features 0x7, context size is
     832 bytes, using 'standard' format.
Sep 21 15:06:35 buster kernel: BIOS-provided physical RAM map:
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff]
     usable
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff]
     reserved
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff]
     reserved
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x0000000000100000-0x00000001ffefffff]
     usable
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000001fff0000-0x00000001ffffffff]
     ACPI data
Sep 21 15:06:35 buster kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff]
     reserved
```

Journalisation - les données de /var/log/syslog

```
root@buster:/home/vagrant# journalctl -f
-- Logs begin at Mon 2020-09-21 15:06:35 GMT. --
Sep 21 19:21:17 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:21:48 buster systemd[1]: Reloading.
Sep 21 19:35:37 buster systemd[1]: Stopping The Apache HTTP
    Server...
Sep 21 19:35:37 buster systemd[1]: apache2.service: Succeeded.
Sep 21 19:35:37 buster systemd[1]: Stopped The Apache HTTP
    Server.
Sep 21 19:35:43 buster systemd[1]: apache2.service: Unit cannot
    be reloaded because it is inactive.
Sep 21 19:35:56 buster systemd[1]: Starting The Apache HTTP
    Server...
Sep 21 19:35:56 buster systemd[1]: Started The Apache HTTP
    Server.
```

Gestion allumage/extinction/redémarrage

```
$ systemctl reboot
$ systemctl poweroff
$ systemctl suspend
```

La gestion des paquets avec apt

Apt

- Outil de gestion standard des paquets au dessus de dpkg pour Debian/Ubuntu
- Gère les dépendances entre paquets $\rightarrow$ évite les installations manuelles
- Permet de nommer les paquets d'après leur nom et non leur numéro de version.
  Ex: libc6 et non libc6_1.9.6-2.deb

Apt

Pourquoi faut-il toujours faire un update?

Pour pointer vers la bonne version dans le dépôt.
Ex: apt croit que libc6 c'est libc6_1.9.5-1.deb et non libc6_1.9.6-2.deb
$\rightarrow$ engendrera une erreur à l'installation

Nota: apt upgrade met à jour tous les paquets existants (mais ne met pas à jour la distribution Debian, par exemple Debian 9 à Debian 10).

Apt: la gestion des sources

Dans le fichier /etc/apt/sources.list (ou dans le répertoire /etc/apt/sources.list.d)

```
root@buster:/home/vagrant# more /etc/apt/sources.list

# deb cdrom:[Debian GNU/Linux 10.0.0 _Buster_ - Official amd64 NETINST
      20190706-10:23]/ buster main
#deb cdrom:[Debian GNU/Linux 10.0.0 _Buster_ - Official amd64 NETINST
      20190706-10:23]/ buster main

# Base
deb http://deb.debian.org/debian buster main
deb-src http://deb.debian.org/debian buster main

# Security
deb http://security.debian.org/debian-security buster/updates main
deb-src http://security.debian.org/debian-security buster/updates main
```

La source peut-être un cdrom, un serveur en http ou https

Apt: les sources

Format :

```
deb url distribution component1 component2 component3 [..] componentX
deb-src url distribution component1 component2 component3 [..] componentX
```

- Les composants sont typiquement : main, contrib, non free
- deb: les binaires
- deb-src: les sources
- les distributions peuvent avoir un type associé : oldstable, stable, testing, unstable. Défaut : stable.
- Les dépôts qui sont au minimum présents sont Base et Security.

Apt: la gestion des sources

- Main les paquets conformes à Debian Free Software Guidelines.
- Non-free paquets non conformes à cette politique, mais qui peuvent être distribués librement.
- Contrib paquets open source mais qui ont besoin d'élements non-free

Apt: Ajout de source externe

1. Il faut assez souvent ajouter une clef pour authenfier le dépot

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

2. Puis ajouter les noms des dépôts dans le fichier sources.list :

```
$ sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/debian \
   $(lsb_release -cs) \
   stable"
```

3. Puis faire un update :

```
root@buster:/home/vagrant# apt update
Hit:1 http://security.debian.org/debian-security buster/updates InRelease
Hit:2 http://deb.debian.org/debian buster InRelease
Get:3 https://download.docker.com/linux/debian buster InRelease [44.4 kB]
Get:4 https://download.docker.com/linux/debian buster/stable Packages [13.3 kB]
Fetched 57.8 kB in 0s (155 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
89 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

4. Puis apt-install docker-ce

R303 - Mesures actives : ping, traceroute, smokeping

Guillaume Urvoy-Keller

September 8, 2022

Ping

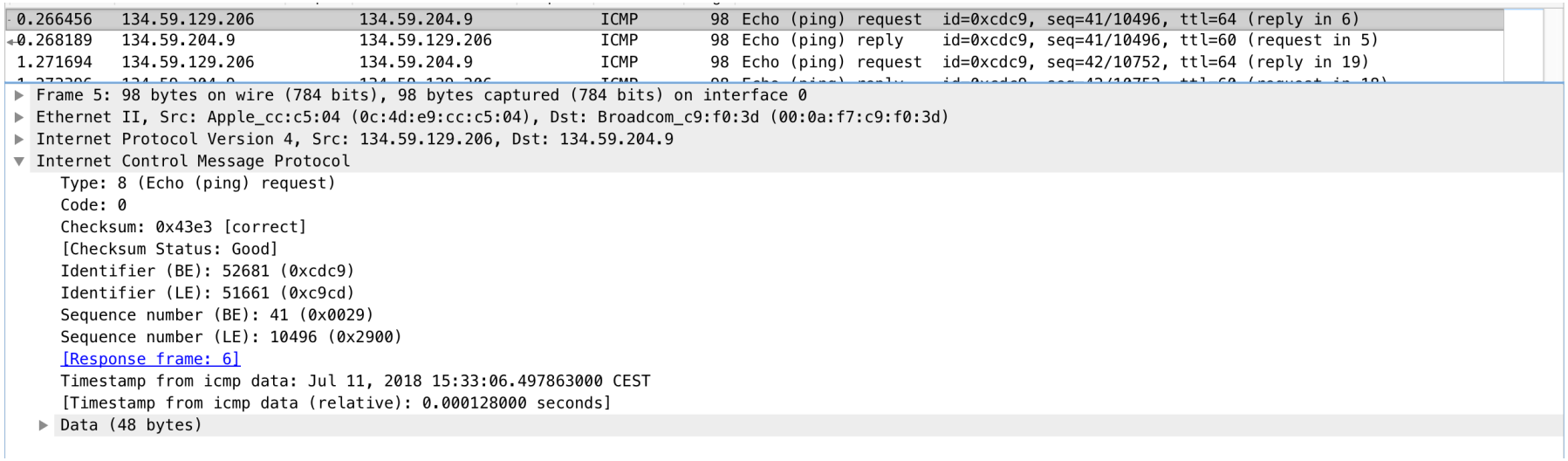
Ping

- Envoie réception de paquet ICMP (Internet Control Message Protocol)
- Code (dans l'en-tête ICMP): echo request et echo reply
- Mesure du temps d'aller-retour (RTT)

Ping

Example

```
guillaumesmbp2:2018 urvoy$ ping www.unice.fr
PING sites.unice.fr (134.59.204.9): 56 data bytes
64 bytes from 134.59.204.9: icmp_seq=0 ttl=60 time=1.810 ms
64 bytes from 134.59.204.9: icmp_seq=1 ttl=60 time=1.714 ms
64 bytes from 134.59.204.9: icmp_seq=2 ttl=60 time=1.864 ms
64 bytes from 134.59.204.9: icmp_seq=3 ttl=60 time=1.932 ms
64 bytes from 134.59.204.9: icmp_seq=4 ttl=60 time=1.936 ms
```

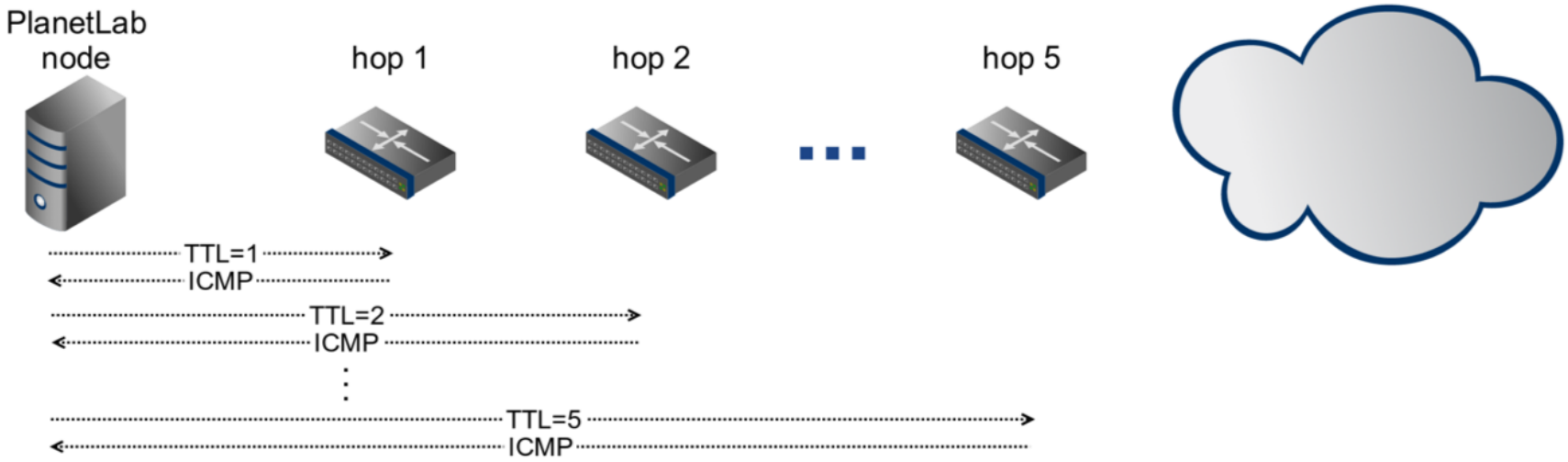


Traceroute

Traceroute

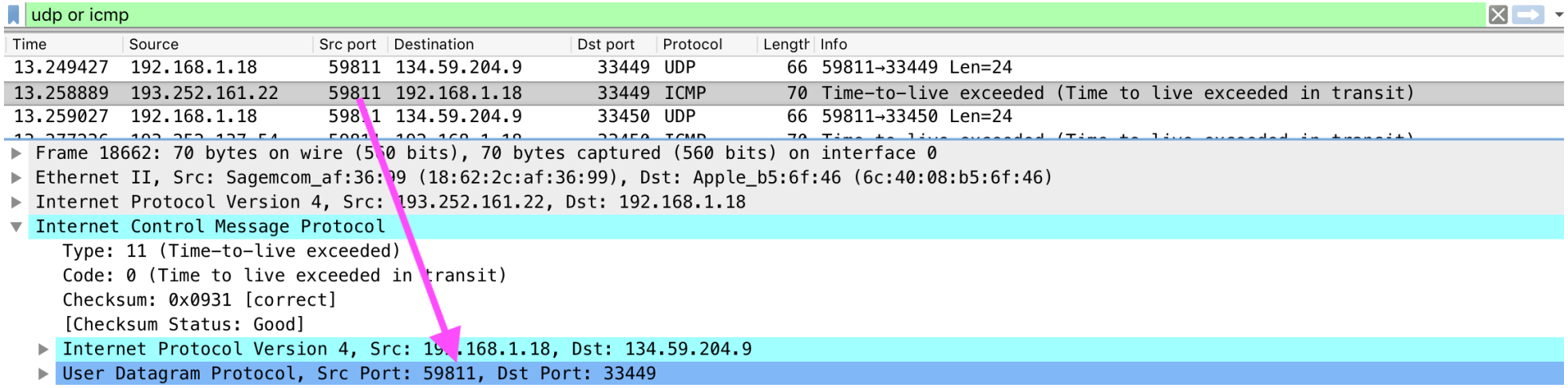
- Traceroute peut envoyer des paquets
  - ICMP
  - UDP ← valeur par défaut
  - TCP

  avec TTL croissant de 1 à 30 (en général)
  et port source destination de 33434 à 33534 ....
  et reçoit des routeurs des paquets ICMP code 11 (TTL expired) + une partie du paquet détruit



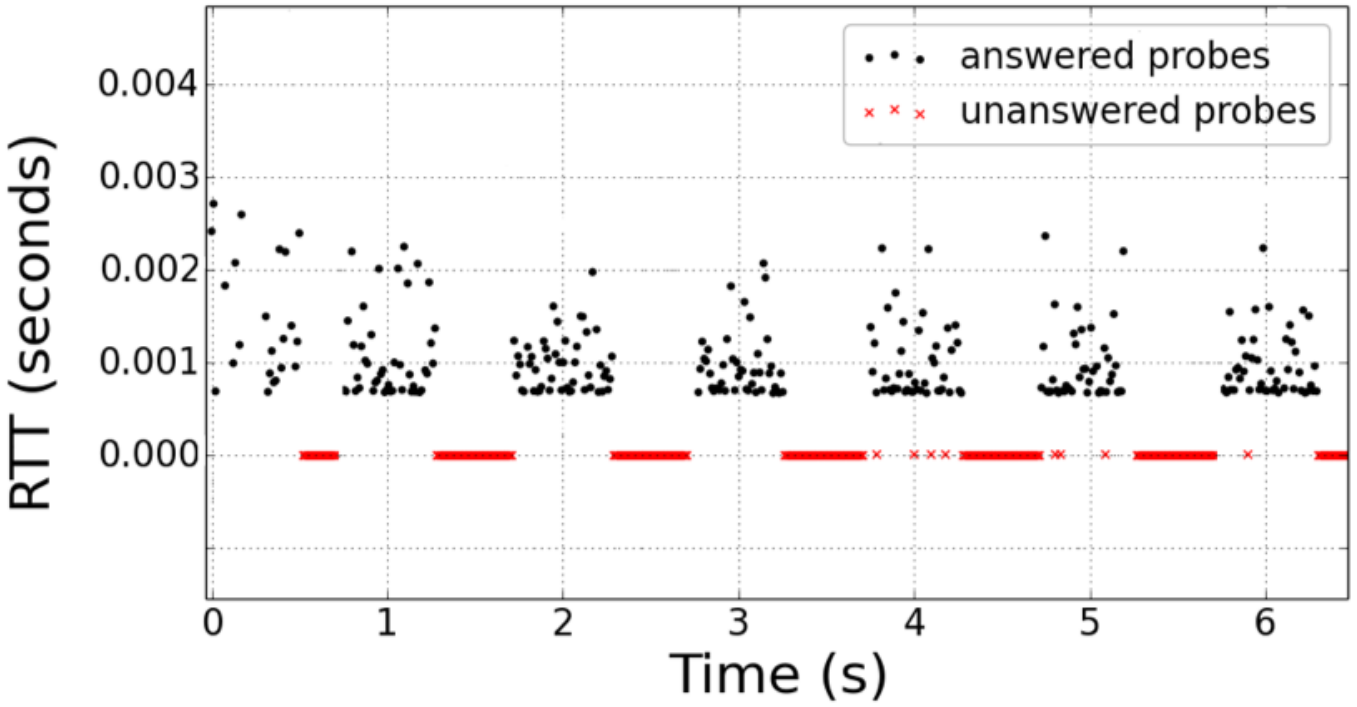
Traceroute

Comportement typique : un paquet UDP envoyé et son en-tête encapsulée dans le paquet ICMP renvoyé par le routeur



Traceroute

- Bonne pratique : Les ISPs doivent configurer leurs routeurs pour répondre à traceroute (répondre en ICMP)
- **mais**, les ISPs veulent prévenir des attaques $\Rightarrow$ limitation du débit de réponse du type x paquets par secondes
- 46 % des routeurs (campagne de mesures 2014-2015, + de 1000 routeurs de l'Internet) :
  - envoie continue de paquets au routeur qui expirent
  - En rouge les paquets sans réponse, en noir avec réponse



Limitation débit de réponse des routeurs aux messages ICMP

Débit fonction de la marque du routeur.
Valeurs typiques : 1 paquet-par-seconde (pps), 10 pps, 50 pps, 100 pps, 500 pps

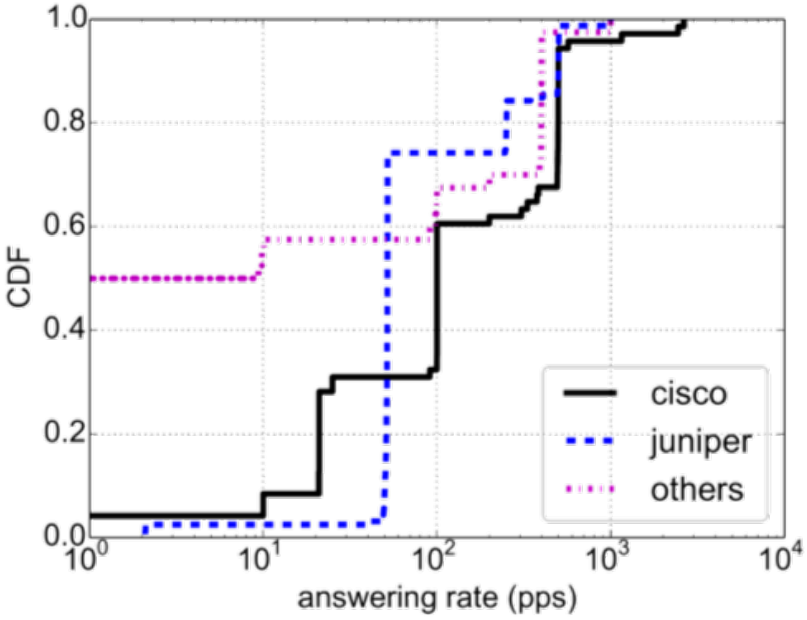


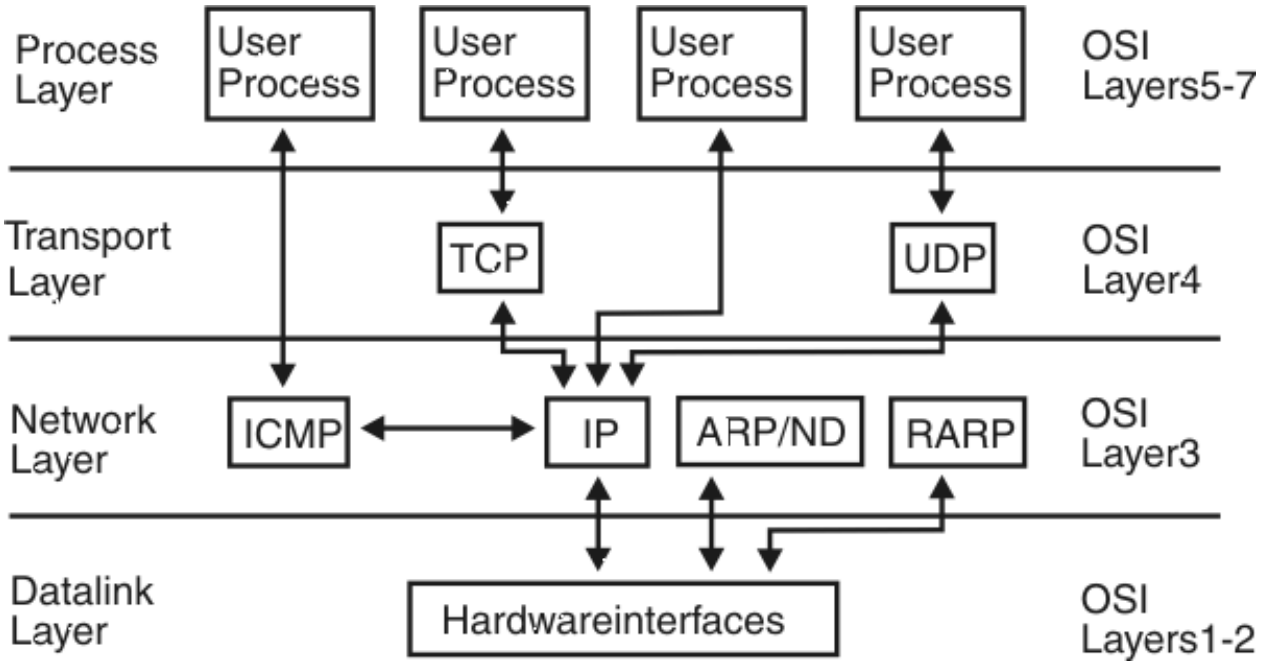
Figure: Riccardo Ravaioli, Guillaume Urvoy-Keller, Chadi Barakat: Characterizing ICMP rate limitation on routers. ICC 2015

Smokeping

Généralisation ping

Pourquoi ne pas faire l'équivalent d'un ping sur différents protocoles ou différentes applications?
Exemples :

- en TCP, on envoie un SYN, on attend un SYN-ACK et on mesure le temps
- en HTTP, on envoie un GET et on attend un 200 OK



Smokeping

- Ping généralisé :
  - « Ping » niveau HTTP $\rightarrow$ un curl sur une page
  - « Ping » niveau SSL $\rightarrow$ openssl sur un serveur
  - « Ping » d'un serveur DNS $\rightarrow$ requête DNS
  - Niveau application : LDAP, NFS
- Représentation graphique avec MRTG
- Base de données RRD (Round Robin Database) :
  - Base de données spécialisées pour stoker des séries temporelles
  - Occupe une taille fixe sur le disque. Les données plus anciennes sont stockées avec une granularité moins fine

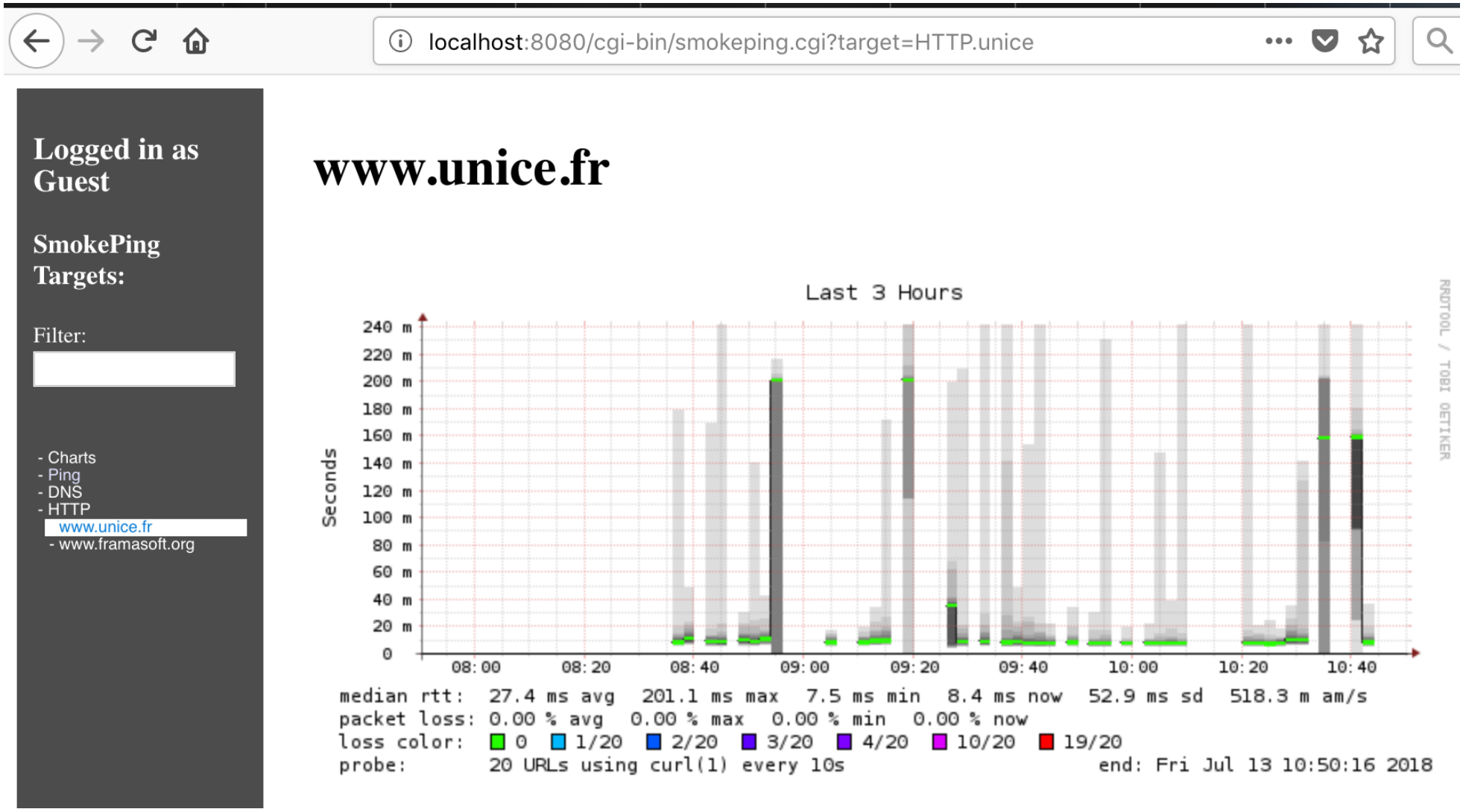
Smokeping

On configure les méthodes de mesure utilisées (Fichier Probes) et les listes de machines sur lesquelles on les applique (Fichier Targets)

```
*** Probes ***

+ FPing
binary = /usr/bin/fping
step = 10

+ DNS
binary = /usr/bin/dig
step = 10

+ Curl
binary = /usr/bin/curl
step = 10
urlformat = http://%host%/

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
Probes                                                    1,1
```

```
*** Targets ***

probe = FPing

menu = Top
title = Network Latency Grapher
remark = Welcome to the SmokePing website of M3105 course.

+ Ping
menu = Ping
title = Ping of local GW of I3S Lab
host = 172.19.130.254

+ DNS
probe= DNS
menu = DNS
title = DNS on 8.8.8.8
host = 8.8.8.8

+ HTTP
menu = HTTP
title = HTTP on unice website and framasoft
probe = Curl

++ unice
host = www.unice.fr

++ framasoft
host= www.framasoft.org

#alerts = someloss
~
~
~
~
~
~
~
~
~
All Targets                                               1,1
```

Smokeping


localhost:8080/cgi-bin/smokeping.cgi?target=HTTP.unice
Logged in as Guest
SmokePing Targets:
Filter:
- Charts
- Ping
- DNS
- HTTP
www.unice.fr
- www.framasoft.org
www.unice.fr
Last 3 Hours
Seconds
240 m
220 m
200 m
180 m
160 m
140 m
120 m
100 m
80 m
60 m
40 m
20 m
0
08:00
08:20
08:40
09:00
09:20
09:40
10:00
10:20
10:40
median rtt: 27.4 ms avg 201.1 ms max 7.5 ms min 8.4 ms now 52.9 ms sd 518.3 m am/s
packet loss: 0.00 % avg 0.00 % max 0.00 % min 0.00 % now
loss color: 0 1/20 2/20 3/20 4/20 10/20 19/20
probe: 20 URLs using curl(1) every 10s
end: Fri Jul 13 10:50:16 2018
RRDTOOL / TOBI OETIKER

R303 - Supervision Système/Application : SNMP

Guillaume Urvoy-Keller

September 8, 2022

Simple Network Management Protocol

- Très ancien protocole
- Pas seulement protocole mais :
  1. Stockage et nomage des données
  2. Un protocole pour y accéder
- Implémenté dans tous les équipements : serveurs, routeurs, imprimantes, points d'accès wifi, caméra IP
- SNMP v1 (1988) n'est plus utilisé et non compatible avec SNMP V2
- SNMP v2 : RFC 1441 (1993)
- SNMP v3 = SNMP v2 + crypto

SNMP

Vocabulaire :

- Agent : service qui tourne dans serveurs, routeurs, imprimantes, points d'accès wifi, caméra IP et maintien base de données
- Manager : service qui interroge l'agent avec des messages *GET* (récupération valeur) et des messages *SET* (écriture d'une valeur, par exemple le nom de l'équipement)
- L'agent peut aussi être configuré pour envoyer des alertes - *trap*
- Protocole SNMP au dessus d'UDP (questions et réponses simples et courtes)

Communication Agent/Manager

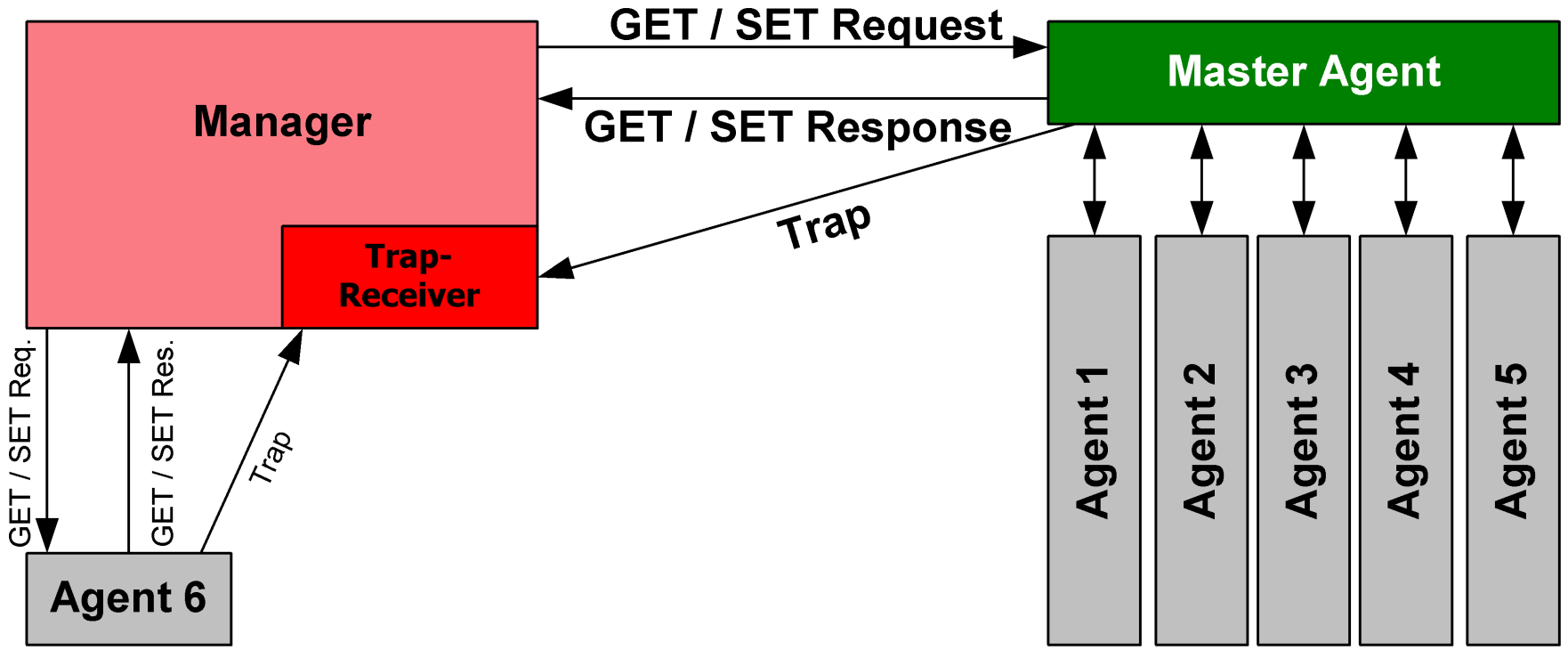


Figure: Source : https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol#/media/File:SNMP_communication_principles_diagram.PNG

MIBs SNMP

- Une base de données d'un agent s'appelle une MIB : Management Interface Base
- En fait, un agent implémente une ou plusieurs MIB :
  - Une MIB de base que l'on trouve partout, la MIB-2
  - des MIBs particulières suivant que l'équipement est un serveur Web, un serveur Radius.

SNMP

Pourquoi SNMP reste intéressant?

1. Parce qu'on le trouve sur tous les équipements réseau.
2. Parce que les bases de données SNMP sont très riches.

Pourquoi est-ce simple d'utilisation (un manager sait lire toutes les données d'un agent si il a les mots de passe)?

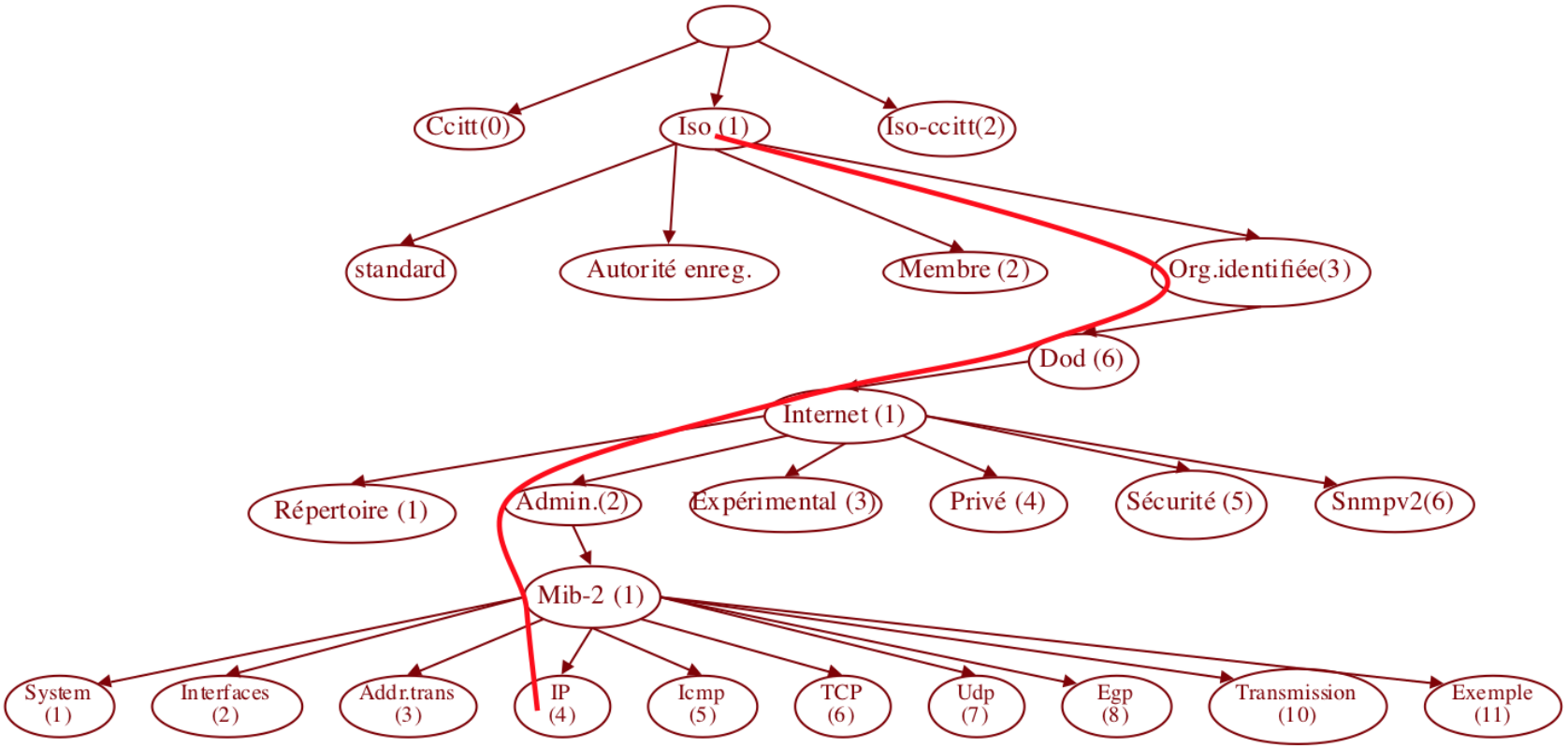
1. Parce que les MIBs sont normalisées et on a une adresse unique dans un grand arbre que tout le monde connait
   - Les objets dans chaque MIB ont un adresse unique $\Rightarrow$ le paramètre sys.Desc qui est une chaine de caractère s'appelle toujours avec un **GET iso.3.6.1.2.1.1.1.0**

Richesse SNMP

Des infos sur le système, les interfaces réseaux, les connexions TCP en cours, le hardware, les processus qui tournent

```
root@ubuntu-xenial:/etc/snmp# snmpwalk -Os -c public -v 2c 127.0.0.1
sysDescr.0 = STRING: Linux ubuntu-xenial 4.4.0-130-generic #156-Ubuntu SMP Thu Jun 14 08:53:28 UTC 2018
sysObjectID.0 = OID: netSnmpAgentOIDs.10
sysUpTimeInstance = Timeticks: (36710) 0:06:07.10
[...]
ifDescr.2 = STRING: Intel Corporation 82540EM Gigabit Ethernet Controller
[...]
tcpConnState.10.0.2.15.22.10.0.2.2.52002 = INTEGER: established(5)
[...]
hrDeviceDescr.196608 = STRING: GenuineIntel: Intel(R) Core(TM) i5-4278U CPU @ 2.60GHz
[...]
hrSWRunName.11581 = STRING: "curl"
hrSWRunName.11630 = STRING: "kworker/1:1"
hrSWRunName.14240 = STRING: "apache2"
hrSWRunName.14243 = STRING: "apache2"
```

Arbre des MIBs



IP = 1.3.6.1.2.1.4

http://www.alvestrand.no/objectid/

http://wwwsnmp.cs.utwente.nl/ietf/mibs/

MIBs d'un agent

- Tout équipement doit implanter la MIB-2, i.e. gérer les objets de cette MIB (RFC 1213) :
  - Statistiques sur les interfaces (vitesse, MTU, octets envoyés/reçus, etc.)
  - Informations sur le système : localisation, contact
- En plus de la MIB-2, un agent implantera :
  - Des MIBs standardisées supplémentaires
    Ex : BGP version 4 MIB (RFC 1657), RADIUS Authentication Server MIB (RFC 2619)
  - Des MIBS propriétaires sous la partie **privé** de l'arbre

Contrôle d'accès SNMP

- Communauté SNMP (v1 et v2) = **mot de passe**
- Une communauté est associée à des droits : read-only, read-write
  - public : read-only
  - private : read-write
- Chaque communauté snmp peut limiter les accès à des objets $\rightarrow$ notion de vues
- Communauté envoyé en clair en SNMP v2
  - Il existe un « authentication trap » pour savoir si quelqu'un a essayé de se connecter avec un mauvais mot de passe
  - On indique en plus l'adresse IP du manager

Contrôle d'accès

Fichier de configuration snmpd.conf d'une machine Debian

```
###############################################################################
#
#  AGENT BEHAVIOUR
#
#  Listen for connections from the local system only
agentAddress  udp:127.0.0.1:161      # Par défaut, agent écoute seulement la loopback
#  Listen for connections on all interfaces (both IPv4 *and* IPv6)
#agentAddress udp:161,udp6:[::1]:161
###############################################################################
#
#  ACCESS CONTROL
#
                                                #  system + hrSystem groups only
view   systemonly  included   .1.3.6.1.2.1.1    # one définit une vue = une partie de l'arbre MIB2
view   systemonly  included   .1.3.6.1.2.1.25.1
                                                #  Full access from the local host
#rocommunity public  localhost
                                                #  Default access to basic system info
 rocommunity public  default    -V systemonly    #on associe une communauté à une vue
                                                #  rocommunity6 is for IPv6
 rocommunity6 public  default   -V systemonly
###############################################################################
#
#  SYSTEM INFORMATION
#
sysLocation    Sitting on the Dock of the Bay # on va faire un SET sur la base de données pour mettre les bonnes valeur
sysContact     Me <me@example.org>
                                                # Application + End-to-End layers
sysServices    72
###############################################################################
#
```

Managers

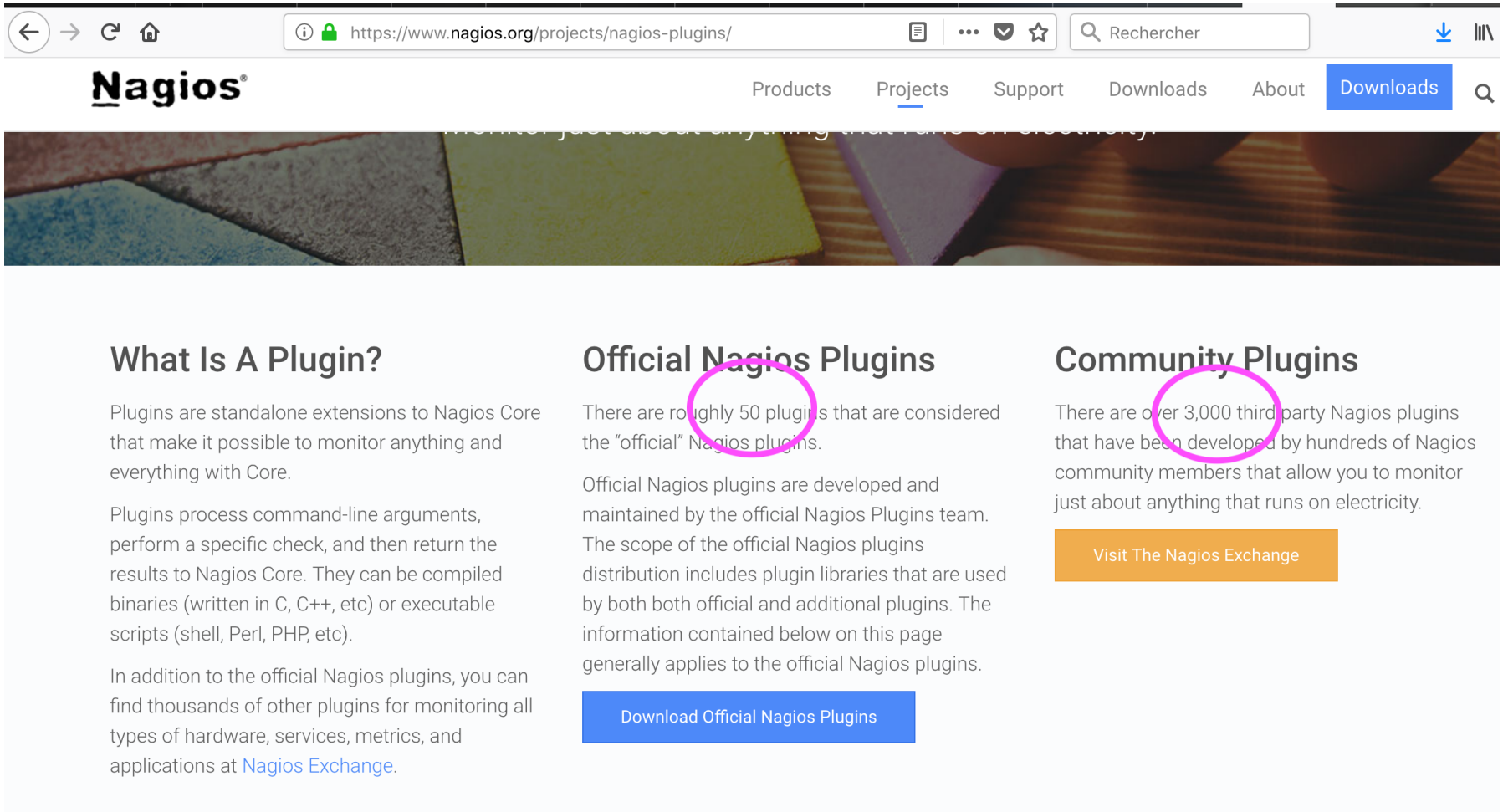
Différentes approches

1. On utilise un outil spécifique
   - Ces outils interrogent en SNMP et font les affichages
   - Ils supportent des pluggins qui s'installent sur les serveurs (en plus des agents SNMP) pour récupérer de données niveau applicatif

   Ex: Nagios, Observium (`http://demo.observium.org`), Cacti, Munim
2. Pour le cas des serveurs, on utilise des suites d'outils qui
   1. collectent les données bas-niveau (SNMP ou niveau applicatif)
   2. les écrivent dans une base de données
   3. Offre un moteur d'affichage pour créer ses propres pages Web

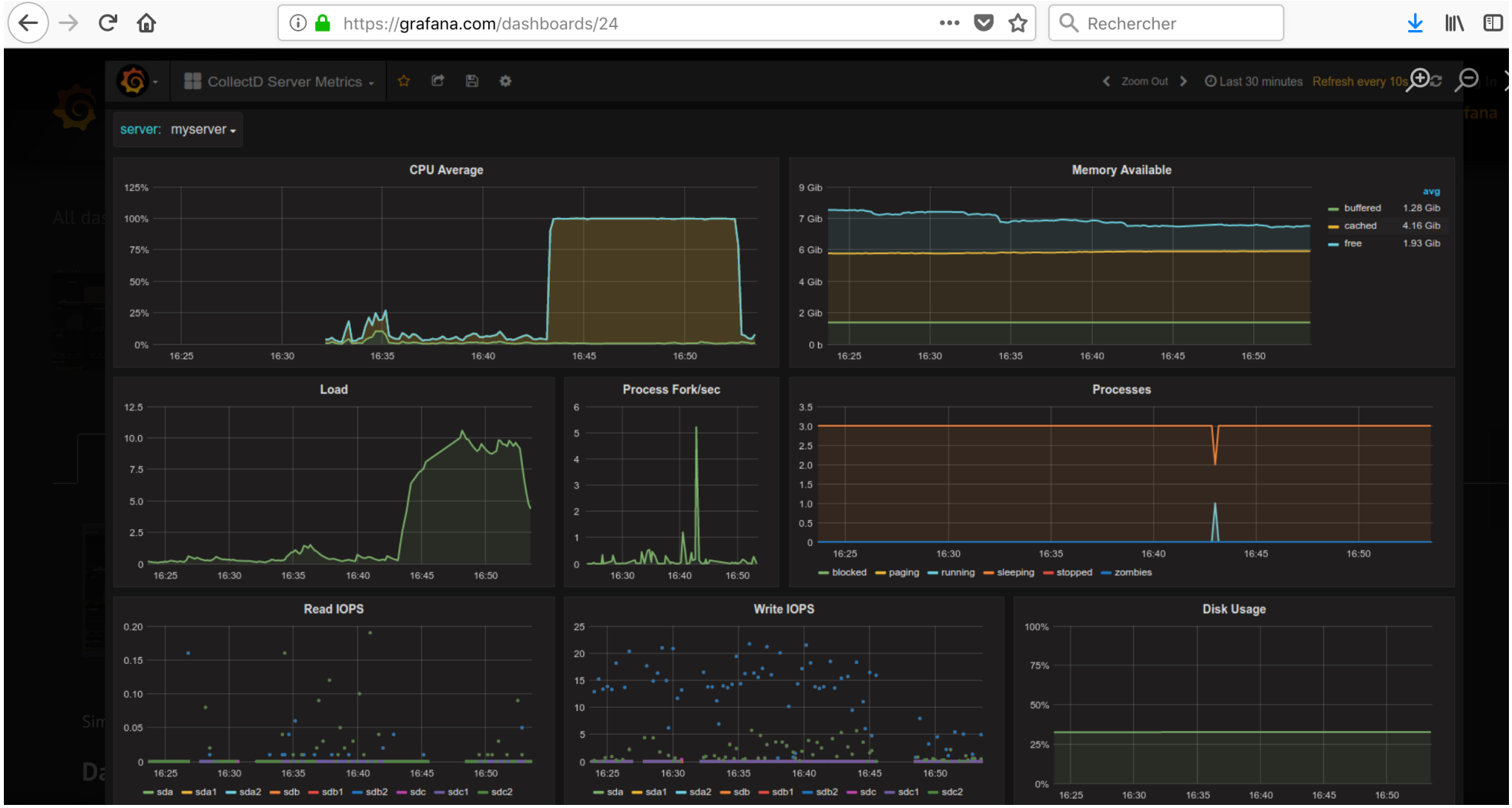
   Ex: Collectd + Graphite + Grafana ou Telegraph

Nagios



Avantage Nagios and co.: fonctionne pour tous les équipements (serveur...caméra IP)

Grafana



Avantage collectd+graphite+grafana : hautement customisables mais réservé aux serveurs.

R303 : Ansible - Configuration automatique de serveurs/services

Guillaume Urvoy-Keller

September 8, 2022

Ansible

Automatisation Configuration

Cycle typique :

1. Installation serveur
2. Configuration $\leftarrow$ Ansible

Configuration

Ce que l'on veut automatiser :

- installation de package
- démarrage/re-démarrage de services
- copies de fichier, par exemple page d'accueil serveur Web.

De plus, on veut :

- pouvoir configurer plein de machines à la fois
  - Ex: 50 machines viennent d'être démarrées sur un cloud public ou privé et il faut les configurer
- appliquer une configuration fonction de la distribution de la machine
  - Ex : *yum install apache2* si Centos et *apt install apache2* si Debian/Ubuntu
- avoir un retour clair sur ce qui a été fait/ce qui a échoué

Ansible et ses concurrents

Les concurrents les plus connus :

- Chef et Puppet
- Puppet
- Ansible

Ansible et ses concurrents

Modèle *Pull* pour Chef/Pupplet

on installe un logiciel sur le serveur à configurer et ce client va interroger un serveur central

Modèle *Push* pour Ansible

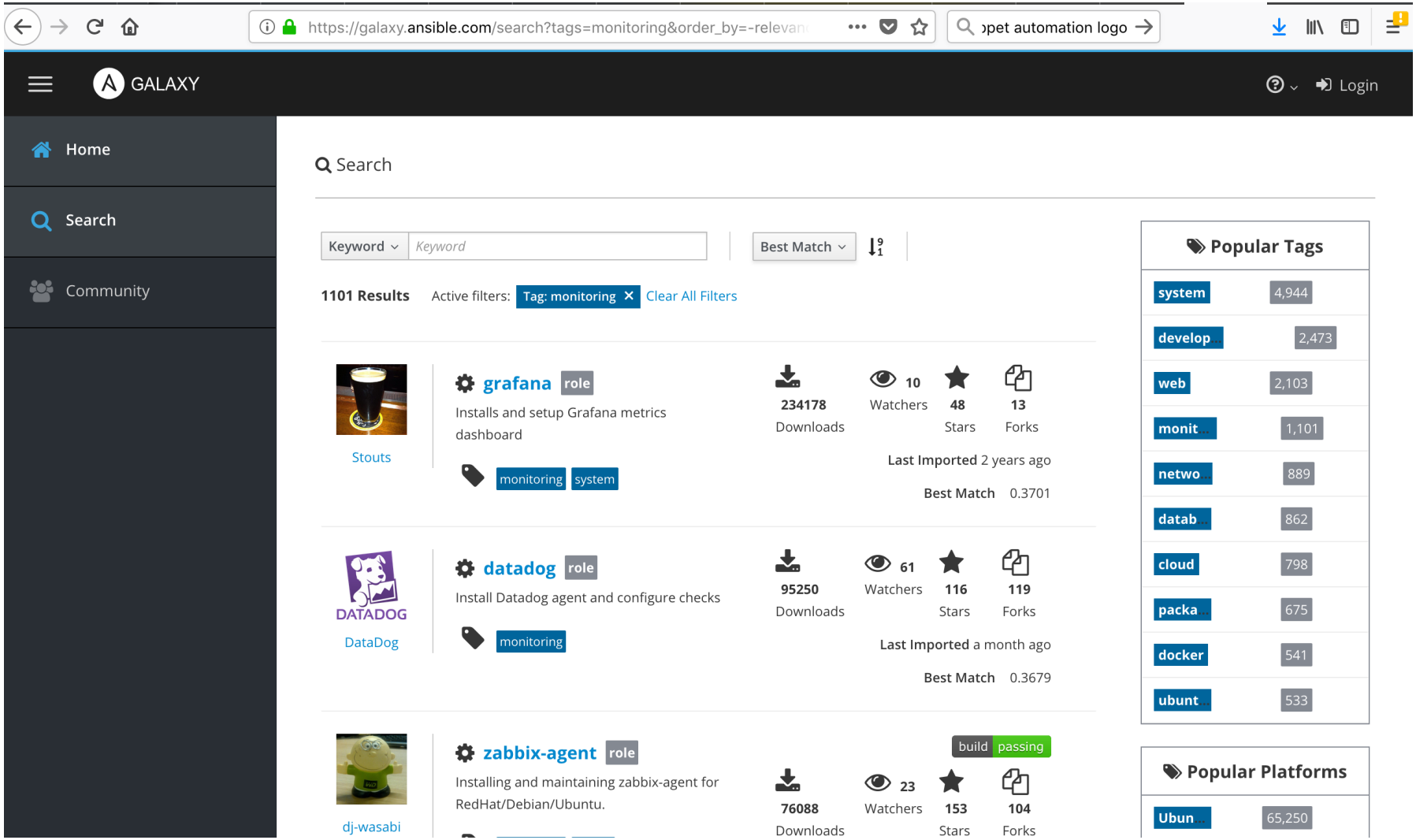
le serveur central pousse directement la configuration en ssh

Ansible - caractéristiques

- Fichiers de configuration simples à écrire et à lire : pas de programmation complexe!
- Notion de **Playbook** : ensemble de commandes à appliquer pour une certaine tâche
  Ex : playbook configuration base serveur centos ou playbook apache
- Notion de **Role** : niveau d'abstraction au dessus d'un playbook
  Ex: permet d'appliquer au serveur centos-apache les rôles *centos* et *apache*.
  Ex: si serveur Ubuntu, alors on appliquera les rôles *ubuntu* et *apache*

Ansible Galaxy

Ansible Galaxy : communauté utilisateurs Ansible s'échange des fichiers de configuration de rôles



Playbook

Ecrits en YAML

```
- name: Configure webserver with nginx
  hosts: testserver
  sudo: True
  tasks:
    - name: install nginx
      apt: name=nginx update_cache=yes
    - name: copy nginx config file
      copy: src=files/default dest=/etc/nginx/sites-available/default
    - name: enable configuration
      file: >
        dest=/etc/nginx/sites-enabled/default
        src=/etc/nginx/sites-available/default
        state=link
    - name: copy index.html
      template: src=templates/index.html.j2 dest=/usr/share/nginx/html/index.html mode=0644
    - name: restart nginx
      service: name=nginx state=restarted
```

Playbook

En-tête : configuration générale
Tâches : liste d'action séquentielle à exécuter sur serveurs, via des modules
Modules : prise en charge d'une action. Ex: modules **apt** ou **service**

Playbook

Variables : au début d'un playbook, Ansible scan la machine cible et récupère des infos (type de distribution, adresse IP, ...).
Ex : module **template va le fichier index.html.j2 et l'instancier**

```
<html>
<head>
<title>
Welcome to ansible
</title> </head>
<body>
<h1>
nginx, configured by Ansible
</h1>
<p>
If you can see this, Ansible successfully installed nginx.
</p>
<p>
{{ ansible_managed }}
</p>
</body>
</html>
```

Ansible Playbook

Dans le playbook, on précise les serveurs cibles.
Le serveur testserveur dans le cas précédent
On peut aussi faire des groupes de serveurs qui seront traités en parallèles. Ex : [web] à la place de testserveur

```
---
[web]
web-1.example.com
web-2.example.com
[db]
db-a.example.com
db-b.example.com
```

Ansible Tower

- Ansible est open-source et supporté par Red-hat
- Red-hat vend un produit qui utilise Ansible : Ansible Tower.
  Ajoute une interface graphique, simplifie encore l'automatisation et le suivi.
  Vise le marché des **devops**