

Architecture Réseaux Entreprises (ARES)

Supervision et visibilité réseau

Brice - Ekane (brice.ekane@univ-rennes.fr)

ISTIC Rennes - France
2025-2026

git clone <https://github.com/bekane/ares-2025.git>

Plan du module

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision
- 3 Outils de Supervision et Architectures
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

Apperçu de la section 1

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision
- 3 Outils de Supervision et Architectures
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

Objectifs pédagogiques avancés

- ▶ Comprendre les impératifs métier et les enjeux techniques de la supervision.
- ▶ Maîtriser le vocabulaire, les modèles et les architectures de supervision.
- ▶ Savoir utiliser et configurer les protocoles clés (**SNMP**, **Syslog**) et les outils (**Nagios**, **Prometheus**, **Grafana**).
- ▶ Développer une méthodologie pour mettre en place une solution de supervision résiliente et pertinente.

La supervision : une nécessité, pas une option

- ▶ **Fiabilité** : Prévenir les pannes et les incidents.
- ▶ **Performance** : S'assurer que les services respectent les S.L.A. (Service Level Agreement).
- ▶ **Sécurité** : Détecter les activités suspectes (sur un port, un flux).
- ▶ **Capacité** : Planifier l'évolution de l'infrastructure (serveurs, bande passante).
- ▶ **L'enjeu** : Transformer des données brutes (métriques, logs) en informations exploitables pour la prise de décision.

Apperçu de la section 2

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision**
- 3 Outils de Supervision et Architectures
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

Le cycle de la supervision

► trois étapes :

- ❶ **Détection et Collecte** : Interroger les équipements ou recevoir des informations de leur part.
- ❷ **Mesure et Analyse** : Comparer les métriques collectées à des seuils définis.
- ❸ **Réaction et Notification** : Déclencher des alertes, envoyer des notifications, ou exécuter des actions correctives.

Types de supervision : active et passive

- ▶ **Supervision active (Polling) :**
 - ▶ **Comment** Le serveur de supervision interroge régulièrement l'équipement (ex : toutes les 5 minutes).
 - ▶ **Avantages** : Contrôle total sur la fréquence, permet de vérifier la disponibilité d'un service.
 - ▶ **Inconvénients** : Charge réseau potentiellement élevée, ne réagit pas aux événements imprévus.
- ▶ **Supervision passive (Traps) :**
 - ▶ **Comment** L'équipement envoie une notification au superviseur uniquement lorsqu'un événement se produit (ex : coupure d'une liaison).
 - ▶ **Avantages** : Faible charge réseau, réactivité instantanée aux événements.
 - ▶ **Inconvénients** : Ne permet pas de vérifier la disponibilité d'un service en l'absence d'événement.

Les métriques clés à collecter

- ▶ **Disponibilité:** L'équipement ou le service est-il joignable ? (ping, port TCP ouvert)
- ▶ **Performances:** Mesure quantitative de la qualité du service.
 - ▶ **Latence :** Délai de réponse (ping, HTTP, etc.).
 - ▶ **Débit :** Bande passante utilisée.
 - ▶ **Perte de paquets :** Pourcentage de paquets perdus.
 - ▶ **Utilisation des ressources :** CPU, RAM, espace disque.

Seuils et états

- ▶ **Pourquoi** Pour transformer une mesure continue en un état binaire ou ternaire.
- ▶ **Comment** On définit des seuils *warning* et *critical*.
- ▶ **Exemple** : Utilisation CPU
 - ▶ OK : $< 70\%$
 - ▶ WARNING : $70\% \leq \text{CPU} < 90\%$
 - ▶ CRITICAL : $\geq 90\%$
- ▶ **États** : La supervision permet de remonter des états, pas seulement des valeurs.
 - ▶ OK : Le service est nominal.
 - ▶ WARNING : Le service fonctionne, mais un problème potentiel est en vue.
 - ▶ CRITICAL : Le service est dégradé ou hors service.
 - ▶ UNKNOWN : L'état n'a pas pu être déterminé.

Apperçu de la section 3

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision
- 3 Outils de Supervision et Architectures**
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

Citation historique

Si jeunesse savait, si vieillesse pouvait

« *Si jeunesse savait, si vieillesse pouvait.* »

— Henri Estienne (1528–1598), Apologies pour Hérodoté (1566)



"C'est vieux, donc ça sert à rien"











- ▶ **Vieux <> Obsolète** : TCP/IP a 40 ans, Linux 30 ans, SQL 50 ans → toujours indispensables.
- ▶ **Ce qui dure, c'est ce qui marche** : une techno répandue vit longtemps dans les entreprises.
- ▶ **Les nouvelles s'inspirent des anciennes** : comprendre Nagios aide à saisir Icinga, Centreon, Prometheus, Zabbix, ...etc
- ▶ **Valeur pro** : beaucoup de "vieilles" technos sont recherchées car moins de gens les maîtrisent.

En bref : connaître les bases anciennes, c'est comme parler le latin pour comprendre toutes les langues modernes.

Nagios : l'outil historique (supervision active)

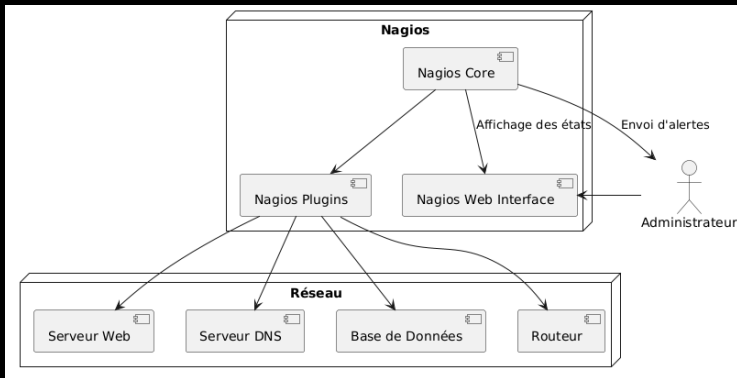
- ▶ Nagios est un moteur de supervision, robuste et très flexible grâce à son architecture de plugins.
- ▶ **Comment ça marche :**
 - ▶ **Le moteur** : Planifie les contrôles, gère les états et envoie des notifications.
 - ▶ **Les plugins** : De petits scripts (Bash, Python, Perl, ...) qui effectuent une vérification et renvoient un code de sortie (0 pour OK, 1 pour Warning, 2 pour Critical, 3 pour Unknown).
 - ▶ **L'architecture** : Basée sur des fichiers de configuration simples pour définir les hôtes et les services.

Companies Currently Using Nagios

Company Name	Website	City	State	Country	Industry	Revenue Range	Employee Range
 Northrop Grumman Corporation	northropgrumman.com	Falls Church	VA	US	Manufacturing	Over \$1,000,000,000	Above 10,000
 Apple, Inc.	apple.com	Cupertino	CA	US	Computer and Electronic Product Manufacturing	Over \$1,000,000,000	Above 10,000
 Raytheon Technologies Corporation	rtx.com	Arlington	VA	US	Manufacturing	Over \$1,000,000,000	Above 10,000
 Comcast Corporation	comcast.com	Philadelphia	PA	US	Media and Entertainment	Over \$1,000,000,000	Above 10,000
 Oracle Corporation	oracle.com	Austin	TX	US	Computer and Electronic Product Manufacturing	Over \$1,000,000,000	Above 10,000
 Dell Technologies Inc.	dell.com	Round Rock	TX	US	Computer and Electronic Product Manufacturing	Over \$1,000,000,000	Above 10,000
 Salesforce, Inc.	salesforce.com	San Francisco	CA	US	Computer and Electronic Product Manufacturing	Over \$1,000,000,000	Above 10,000
 Cisco Systems, Inc.	cisco.com	San Jose	CA	US	Computer and Electronic Product Manufacturing	Over \$1,000,000,000	Above 10,000
 AT&T Inc.	att.com	Dallas	TX	US	Telecommunications	Over \$1,000,000,000	Above 10,000
 Verizon Communications Inc.	verizon.com	New York	NY	US	Telecommunications	Over \$1,000,000,000	Above 10,000

Nagios

Architecture et fonctionnement



Architecture et fonctionnement

- ▶ Nagios Core : moteur de supervision central.
- ▶ Hôtes (Hosts) : systèmes à surveiller (serveurs, routeurs).
- ▶ Services : services exécutés sur les hôtes (HTTP, DNS, etc.).
- ▶ Plugins : scripts de vérification (OK, Warning, Critical).
- ▶ Notifications : alertes par e-mail ou SMS en cas de problème.

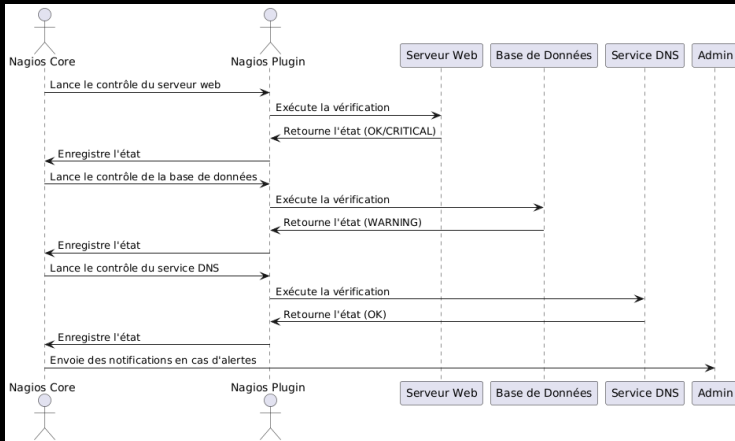
Fonctionnement de Nagios

Étapes principales

- ▶ Planification des contrôles : vérifications régulières des services.
- ▶ Collecte des données via les plugins.
- ▶ Gestion des événements : déclenchement des alertes ou des actions correctives.

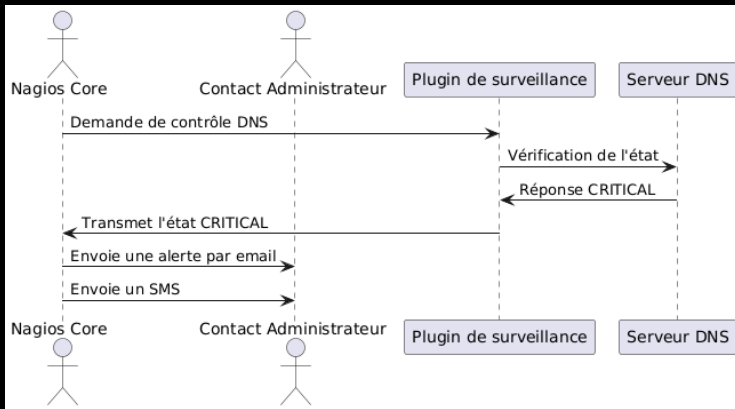
Fonctionnement de Nagios

Déroulement d'un plan de controle



Fonctionnement de Nagios

Déroulement d'une notification



Configuration des services et hôtes

Principes

- ▶ Utilisation de fichiers de configuration.
- ▶ Exemple : surveillance d'un serveur web avec l'adresse IP 192.168.1.10.
- ▶ Vérification de services critiques : HTTP, DNS, base de données.

Configuration des services et hôtes

Définition d'un hôte (Host) :

```
define host {  
    use                linux-server  
    host_name          webserver1  
    alias              Web Server 1  
    address            192.168.1.10  
}
```

Configuration des services et hôtes

Définition d'un service

```
define service {  
    use                generic-service  
    host_name          webserver1  
    service_description HTTP Service  
    check_command       check_http  
}
```

Création d'alertes et tableau de bord

Exemple de configuration d'une notification

```
define contact {  
    contact_name          admin  
    email                 admin@company.com  
    service_notification_commands notify-service-by-email  
    host_notification_commands notify-host-by-email  
}
```

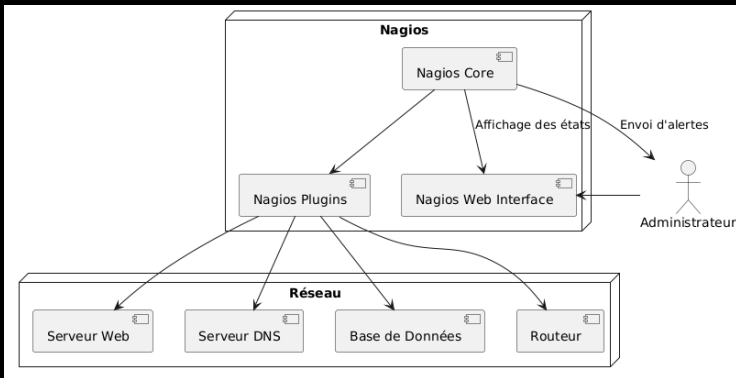

Cas pratique : Déploiement de Nagios

Étapes principales

- ▶ Installation de Nagios Core et des plugins.
- ▶ Configuration des hôtes et services à surveiller.
- ▶ Mise en place des alertes et des contacts pour notifications.
- ▶ Accès à l'interface web et suivi des services.

Cas pratique : Déploiement de Nagios

Exemple de déploiement



Add-on

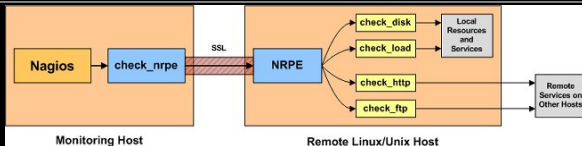
Liens

- ▶ Nagios Cross Platform Agent (NCPA)
- ▶ Nagios Remote Plugin Executor (NRPE)
- ▶ Nagios Remote Data Processor (NRDP)

NRPE — Apperçu

Utilisation principale

- ▶ **NRPE** est utilisé pour superviser des ressources locales ou privées d'une machine Linux/Unix distante.
- ▶ Typiquement, il s'agit de mesures qui ne sont pas accessibles directement depuis l'hôte de supervision.

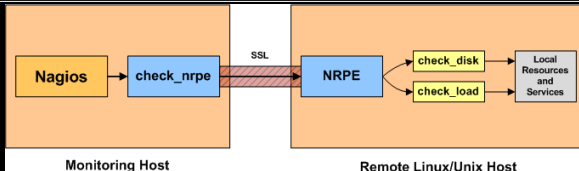


source : <https://support.nagios.com/kb/article/nrpe-architecture-141.html>

NRPE — Supervision des ressources locales

Utilisation principale

- ▶ **NRPE** est utilisé pour superviser des ressources locales ou privées d'une machine Linux/Unix distante.
- ▶ Typiquement, il s'agit de mesures qui ne sont pas accessibles directement depuis l'hôte de supervision.

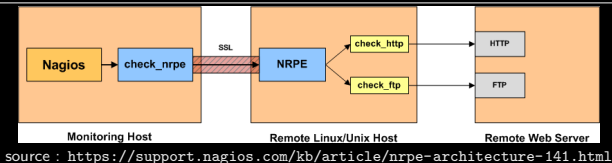


source : <https://support.nagios.com/kb/article/nrpe-architecture-141.html>

NRPE comme proxy de supervision

Idée clé

- ▶ **NRPE** peut servir à superviser indirectement des services publics.
- ▶ Cas d'usage : le serveur distant avec NRPE peut accéder à une ressource que l'hôte de supervision ne peut pas atteindre.
- ▶ Le démon NRPE agit alors comme **proxy de supervision**.



Solutions Comerciales

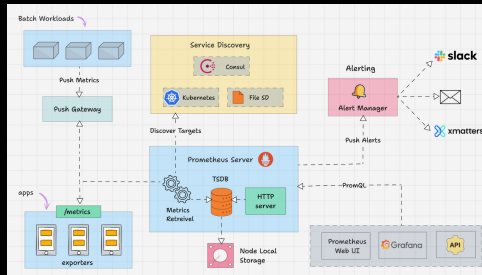
Liens

- ▶ Nagios XI
- ▶ Nagios Log Server
- ▶ Nagios Network Analyzer
- ▶ Nagios Fusion

Prometheus : l'approche Cloud Native

- ▶ Réponse à une architecture moderne, dynamique et distribuée (containers, microservices).
- ▶ **Comment ça marche** :
 - ▶ **Le modèle "Pull"** : Le serveur Prometheus "scrappe" (va chercher) les métriques sur les cibles à intervalles réguliers.
 - ▶ **Exporters** : Les cibles exposent leurs métriques via un serveur HTTP.
 - ▶ **Base de données temporelle** : Stockage optimisé des métriques avec leur timestamp.
 - ▶ **PromQL** : Un langage de requête puissant pour l'analyse des métriques.

Prometheus : architecture

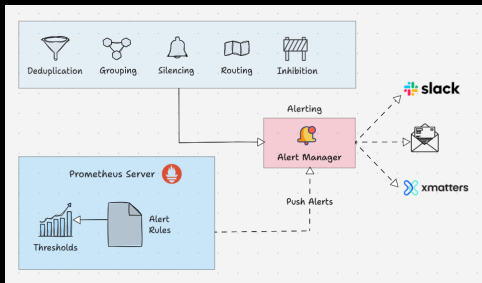


source : <https://devopscube.com/prometheus-architecture/>

Prometheus Server

- ▶ **Cerveau du monitoring** : collecte toutes les métriques.
- ▶ **Modèle pull** : interroge les cibles (apps, workloads, cluster kubernetes).
- ▶ **Scraping** : processus de collecte périodique des métriques.

Prometheus Alertmanager



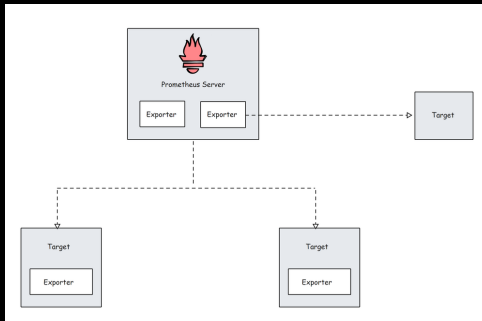
source : <https://devopscube.com/prometheus-architecture/>

- ▶ **Gestion des alertes** générées à partir des règles Prometheus.
- ▶ **Déduplication et regroupement** des alertes similaires.
- ▶ **Notifications** vers e-mail, Slack, xmateters, etc.

Prometheus TSDB

- ▶ **Stockage des métriques** sous forme de séries temporelles.
- ▶ **Éléments d'une métrique** : nom, labels, valeur, timestamp.
- ▶ **Optimisé pour PromQL** (requêtes et analyses de métriques).

Prometheus : Cible/Target

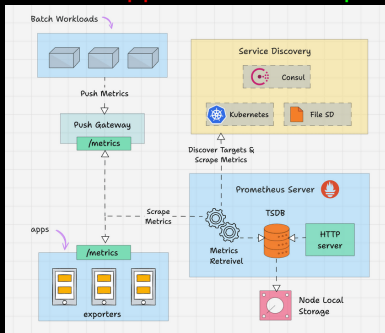


source : <https://devopscube.com/prometheus-architecture/>

- **Source des métriques** que Prometheus vient scraper.
- Peut être un serveur, un service, un pod Kubernetes ou un endpoint applicatif.

Prometheus : scrapping

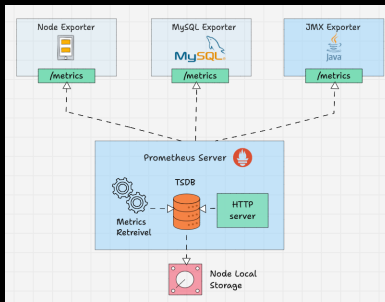
Prometheus ne reçoit pas les métriques, il va les chercher lui-même en interrogeant régulièrement les applications. **Scrape Metrics**



source : <https://devopscube.com/prometheus-architecture/>

- Chaque application (ou exporter) expose ses métriques sur une URL **/metrics**.

Prometheus : exporters

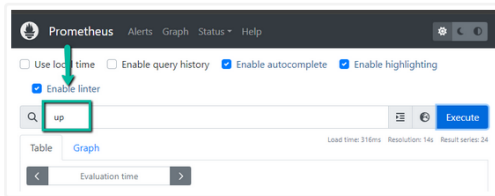


source : <https://devopscube.com/prometheus-architecture/>

- ▶ **Exporter = agent** : expose les métriques au format Prometheus via /metrics.
- ▶ **Metrics Retrieval** : Prometheus collecte et structure ces données.
- ▶ **Écosystème** : nombreux exporters (officiels ou tiers), extensibles.

Prometheus : promptql

Prometheus UI



Query over the CLI

```
curl "http://54.186.154.78:30000/api/v1/query?query=$(echo 'up'
```

source :

<https://devopscube.com/prometheus-architecture/>

Grafana : visualisation et tableaux de bord

- ▶ Pour transformer des données brutes en visualisations compréhensibles et dynamiques.
- ▶ **Comment ça marche** :
 - ▶ **Data Sources** : Se connecte à différentes sources de données (Prometheus, Nagios, SQL, etc.).
 - ▶ **Tableaux de bord (Dashboards)** : Permet de créer des vues personnalisées avec des graphes, des jauges, des tables, etc.
 - ▶ **Alerting** : Permet de déclencher des alertes basées sur des requêtes complexes, plus flexibles que les seuils simples de Nagios.

Ressources recommandées — Grafana et Prometheus

Documentation officielle :

- ▶ Grafana — Getting Started :
[https://grafana.com/docs/grafana/latest/getting-started/:contentReference\[oaicite:0\]index=0](https://grafana.com/docs/grafana/latest/getting-started/:contentReference[oaicite:0]index=0)
- ▶ Prometheus — Getting Started Tutorial :
[https://prometheus.io/docs/tutorials/getting_started/:contentReference\[oaicite:1\]index=1](https://prometheus.io/docs/tutorials/getting_started/:contentReference[oaicite:1]index=1)

Tutoriels francophones :

- ▶ Xavki — “Prometheus/Grafana : tutoriels français” (guide pratique en français) :
[https://xavki.blog/prometheus-grafana-tutoriaux-francais/:contentReference\[oaicite:2\]index=2](https://xavki.blog/prometheus-grafana-tutoriaux-francais/:contentReference[oaicite:2]index=2)

Apperçu de la section 4

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision
- 3 Outils de Supervision et Architectures
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

SNMP : Simple Network Management Protocol

- ▶ **SNMP** C'est le standard de facto pour la gestion et la supervision d'équipements hétérogènes.
- ▶ **Un modèle client-serveur** :
 - ▶ Un **Manager** (le superviseur) qui envoie des requêtes.
 - ▶ Un **Agent** (l'équipement) qui répond aux requêtes et gère une base de données d'informations.
- ▶ **Le problème du broadcast** : Les requêtes ne sont pas en broadcast, mais en unicast. Le Manager doit connaître l'adresse de l'Agent.

SNMP : Protocoles et ports

- ▶ Fonctionne principalement sur UDP.
- ▶ Port 161 : requêtes/commandes.
- ▶ Port 162 : TRAPs et INFORMs.
- ▶ Extensions possibles : SNMP sur TLS/DTLS.

SNMP : Fonctionnement et requêtes

- ▶ **Polling** : GET, GET NEXT, GET BULK, SET.
- ▶ **Notifications** : TRAP et INFORM (alertes envoyées par l'agent).
- ▶ Permet supervision active et réactive.

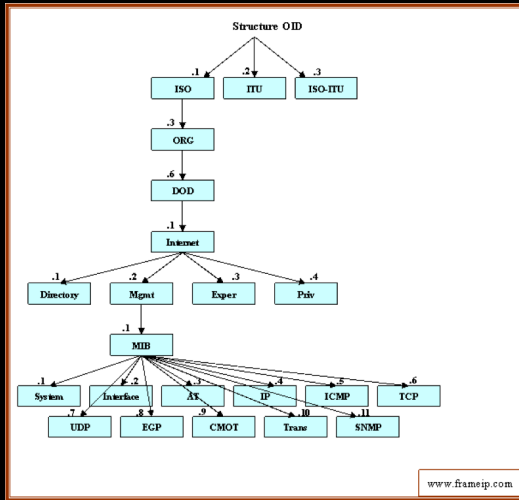
SNMP : Configuration typique

- ▶ Activer l'agent SNMP sur Linux (snmpd) ou Windows.
- ▶ Définir les communautés (v1/v2) ou utilisateurs (v3).
- ▶ Adapter la configuration : `/etc/snmp/snmpd.conf`.

La MIB (Management Information Base)

- ▶ Une base de données hiérarchique d'objets, qui définit les informations que l'on peut lire sur un équipement.
- ▶ **L'OID (Object Identifier)** : Un chemin unique pour chaque objet dans la MIB.
- ▶ Garantir l'interopérabilité. Un OID est le même pour un routeur Cisco et un switch Juniper.
- ▶ **Comment ça marche ?** Les OID sont représentés par des nombres (ex : .1.3.6.1.2.1.1.1.0 pour la description du système).

La MIB (Management Information Base)



Versions de SNMP

▶ SNMPv1 et SNMPv2c :

- ▶ **Comment ?** Utilisation d'une chaîne de caractère ("community string") pour l'authentification.
- ▶ **Inconvénients** : La chaîne est transmise en clair, aucune confidentialité ni intégrité des données. Très dangereux sur un réseau non-fiable.

▶ SNMPv3 :

- ▶ **Comment ?** Ajoute la confidentialité (chiffrement), l'intégrité (hachage) et l'authentification forte.
- ▶ **Avantages** : Standard moderne et sécurisé, indispensable pour la supervision sur Internet.

(Traps) SNMP

- ▶ Pour que l'équipement puisse avertir le superviseur d'un événement sans attendre d'être interrogé.
- ▶ **Comment** L'Agent SNMP envoie un message (un "trap") au Manager SNMP pour signaler un événement spécifique.
- ▶ **Exemples de traps :**
 - ▶ Redémarrage de l'équipement (coldStart).
 - ▶ Coupure d'une interface réseau (linkDown).
 - ▶ Utilisation CPU excessive.

Outils en ligne de commande SNMP

```
# snmpget : obtenir une valeur pour un OID donné
# -c community-string, -v version
snmpget -v 2c -c public 192.168.1.1 .1.3.6.1.2.1.1.1.0

# snmpwalk : parcourir une partie de la MIB
# -v version, -c community-string
snmpwalk -v 2c -c public 192.168.1.1 .1.3.6.1.2.1.2.2.1.2
```

Syslog : la journalisation centralisée

► Pourquoi centraliser les logs ?

- Pour l'audit et la traçabilité des événements.
- Pour la corrélation d'événements.
- Pour la détection d'intrusions (IDS).
- Pour simplifier la gestion des logs de multiples serveurs.

Comment Syslog fonctionne ?

- ▶ **Le daemon Syslog** : 'rsyslogd' (Linux) ou 'syslog-ng'.
- ▶ **Le protocole** : Envoie les messages via UDP ou TCP sur le port 514.
- ▶ **Les composants d'un message** :
 - ▶ **Facility** : Catégorie du message (kernel, mail, auth, etc.).
 - ▶ **Severity** : Niveau d'urgence (emerg, alert, crit, err, warning, info, debug).
- ▶ **Agrégation** : Collecter les logs sur un serveur centralisé.

Syslog: format de message

BSD-syslog Format (RFC 3164, puis RFC 5424)

Syslog: exemple de configuration

Fichier de configuration (/etc/rsyslog.conf ou /etc/rsyslog.d/*.conf)

```
1  # Journal noyau
2  kern.*                               /var/log/kern.log
3
4  # Authentification
5  auth,authpriv.*                     /var/log/auth.log
6
7  # Messages de mail (séparation en fonction du niveau)
8  mail.info                           /var/log/mail.info
9  mail.warn                           /var/log/mail.warn
10 mail.err                            /var/log/mail.err
11
12 # Tous les logs sauf mail
13 *.info;mail.none;authpriv.none     /var/log/messages
14
15 # Rediriger tous les logs vers un serveur syslog central
16 *.*                                 @192.168.1.100:514
17
```

Actions possibles de syslog

Exemples de destinations

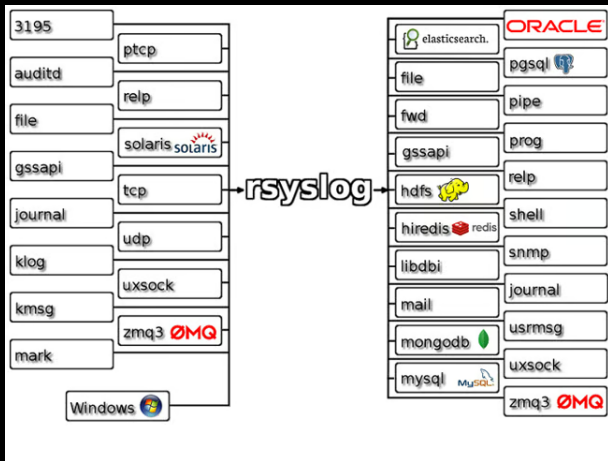
- ▶ **Fichier local** : `/var/log/secure`
- ▶ **Terminal/console** : `/dev/tty10`
- ▶ **Serveur distant UDP** : `@192.168.1.100`
- ▶ **Serveur distant TCP (sécurisé)** : `@@logs.example.com:514`
- ▶ **Script** : `||usr/local/bin/myscript.sh`

Syslog : Configurer la réception distante

Dans `/etc/rsyslog.conf` (ou `/etc/rsyslog.d/remote.conf`) :

```
1  # Activer la réception TCP et UDP
2  module(load="imudp")
3  input(type="imudp" port="514")
4
5  module(load="imtcp")
6  input(type="imtcp" port="514")
7
```

Syslog : Exemple de déploiement



source : <https://betterstack.com/community/guides/logging/rsyslog-explained/>

NetFlow et sFlow : l'analyse de flux

- ▶ utilisation détaillée du réseau, au-delà du simple nombre de paquets.
- ▶ **Comment ça marche ?** :
 - ① Le routeur/switch identifie les flux (paires d'adresses IP, ports, protocole).
 - ② Il agrège les informations sur ces flux et les exporte vers un collecteur.
 - ③ Le collecteur centralise les données pour analyse.
- ▶ **NetFlow (Cisco)** : Échantillonnage ou non.
- ▶ **sFlow (standard)** : Échantillonnage obligatoire, très léger pour le processeur de l'équipement.

Monitoring avec sFlow

Principe

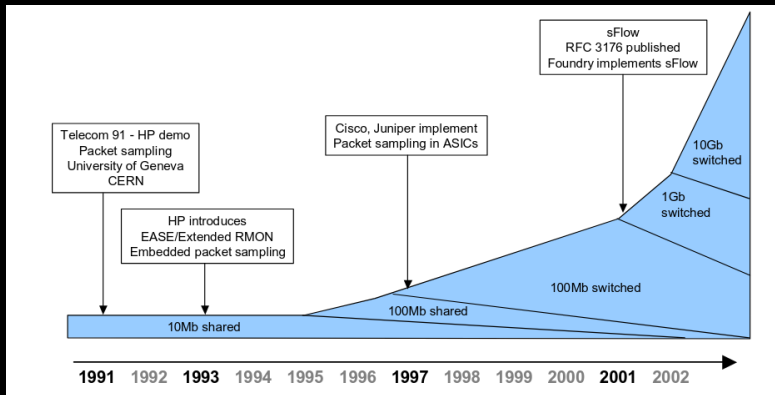
- ▶ Les **agents sFlow**, intégrés aux équipements réseau, échantillonnent le trafic et collectent des compteurs.
- ▶ Les échantillons et statistiques sont envoyés vers un **collecteur sFlow**.
- ▶ Le collecteur agrège, analyse et fournit une **vue temps réel du trafic**.

sFlow : Metrics

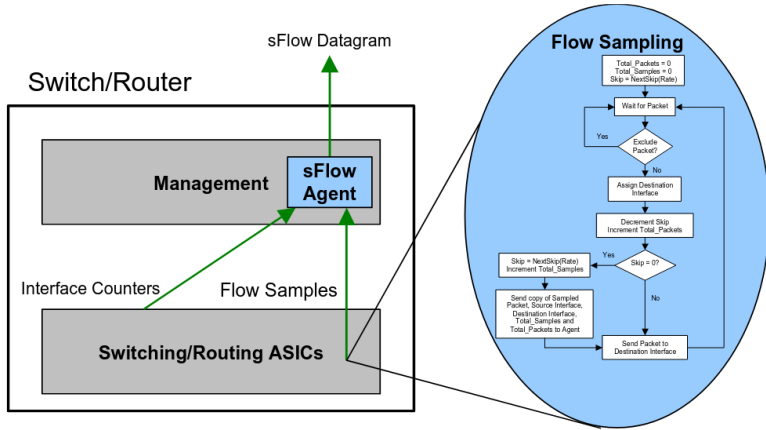
Ce que l'on peut observer

- ▶ Débits sur les interfaces et charge réseau.
- ▶ Types de trafic (protocoles, applications).
- ▶ Flux dominants (top talkers).
- ▶ Anomalies et congestions.

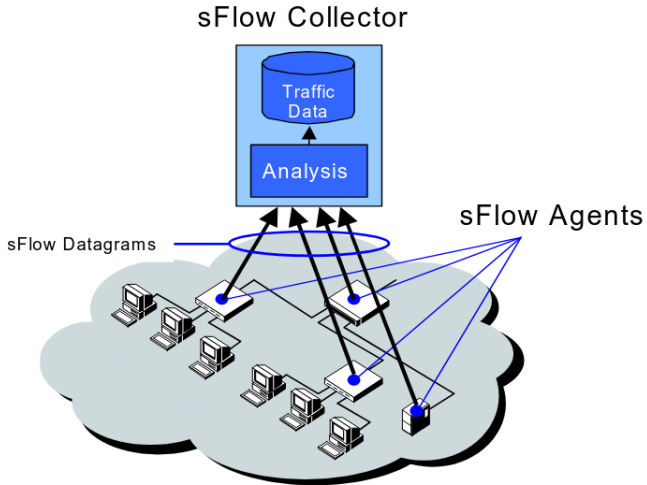
sFlow: Evolution



sFlow: Architecture



sFlow: Architecture



Exemple de collecteur sFlow

Outils disponibles

- ▶ **sFlowTrend** : interface graphique gratuite (InMon)
- ▶ **Host-sflow** : collecte + export vers InfluxDB/Prometheus
- ▶ **ntopng** : analyse réseau en temps réel (supporte sFlow)

Host-sflow: configuration

- ▶ Démon hsflowd
- ▶ fichier de configuration /etc/hsflowd.conf

Configuration simple (hsflowd.conf)

```
sflow {  
  collector {  
    ip = 0.0.0.0  
    udpport = 6343  
  }  
  logfile = /var/log/hsflowd.log  
}
```

Références

- ▶ ExtraHop. **Configuration de NetFlow**. <https://docs.extrahop.com/fr/25.3/configure-netflow/configure-netflow.fr.pdf>
- ▶ Allied Telesis. **sFlow Feature Overview Guide**. https://www.alliedtelesis.com/sites/default/files/documents/configuration-guides/sflow_feature_overview_guide.pdf
- ▶ InMon Corp. **sFlow Overview**. <https://sflow.org/sFlowOverview.pdf>

Apperçu de la section 5

- 1 Introduction et Objectifs du Module
- 2 Principes Fondamentaux de la Supervision
- 3 Outils de Supervision et Architectures
- 4 Protocoles et Technologies Clés
- 5 Stratégie de Supervision

Élaborer une stratégie de supervision

- ▶ **Quoi superviser** Définir les éléments critiques (routeurs, firewalls, serveurs d'application).
- ▶ **Qui est responsable** Définir un plan de réaction et des rôles.
- ▶ **Comment superviser** Choisir les bons outils et protocoles.
- ▶ **Comment notifier** Définir des canaux de notification (email, SMS, outils de collaboration).

Conclusion - Points clés

- ▶ La supervision est **indispensable** pour garantir fiabilité, performance et sécurité.
- ▶ Elle repose sur des **principes fondamentaux** : collecte, analyse, réaction.
- ▶ Les **protocoles historiques** (SNMP, Syslog, NetFlow/sFlow) et les **outils modernes** (Nagios, Prometheus, Grafana) sont complémentaires.
- ▶ La valeur ajoutée vient de la capacité à **transformer des données brutes en décisions**.

Conclusion - Ouverture

- ▶ Savoir superviser, c'est aussi savoir **anticiper et dimensionner**.
- ▶ Les tendances actuelles : **observabilité, cloud native**, intégration avec l'IA.
- ▶ Prochaine étape : approfondir la **virtualisation réseau** et l'intégration avec **Open vSwitch, Libvirt et l'automatisation**.