Atelier de professionnalisation : Initiation à PRTG

Adrien BLAIZE

SOMMAIRE

1. Supervision des Services Web (LAMP)

1.1 Mise en œuvre du capteur HTTP

Ce capteur a pour but de surveiller l'état de ma page WEB "siteab" que j'ai créé et configuré sur mon serveur L.A.M.P en 172.30.206.80.

```
cd /var/www/html/
mkdir siteab
cd siteab
touch index.html
nano index.html
```

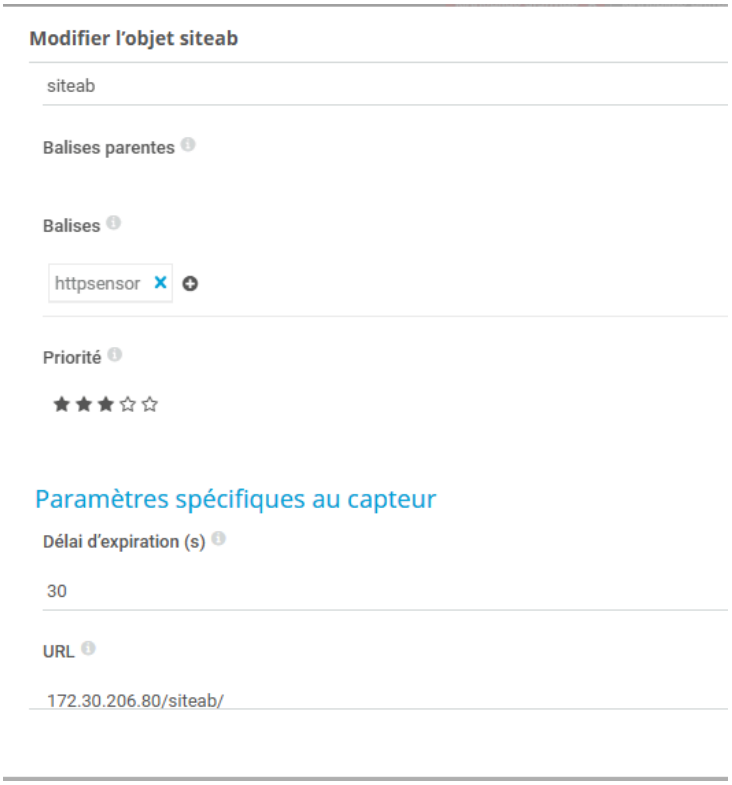
Je me positionne dans le répertoire racine du serveur web (`/var/www/html/`), je crée un dossier spécifique nommé `siteab`, puis j'y génère un fichier `index.html` vide que j'ouvre avec l'éditeur de texte **nano** sur mon serveur LAMP en 172.30.206.80.

La deuxième capture montre l'édition du fichier où j'insère simplement la balise `<h1>siteAB OK</h1>`.

En tapant l'adresse IP `172.30.206.80/siteab/`, je constate que le message "siteAB OK" s'affiche correctement. Cela valide que mon service Apache fonctionne et que le chemin d'accès est bien configuré avant de passer à la supervision.

1.2 Installation du capteur HTTP

Une fois le capteur créé, je procède à sa configuration dans l'interface de supervision.



Je nomme l'objet `siteab` et je lui attribue une priorité de 3 étoiles pour qu'il soit bien visible dans mon tableau de bord.

Je définis un **délai d'expiration de 30 secondes** (timeout). C'est le temps maximum que le capteur attendra avant de considérer que la page ne répond pas.
Je renseigne l'URL précise de mon serveur : `172.30.206.80/siteab/`.

1.3 Arrêt du service Apache2 & Résultat de la supervision

Cet arrêt permet notamment au capteur HTTP V2 d'observer l'état d'arrêt de ma page WEB et de l'inclure en rouge sur PRTG.
Ainsi, on à une démonstration claire de son efficacité.

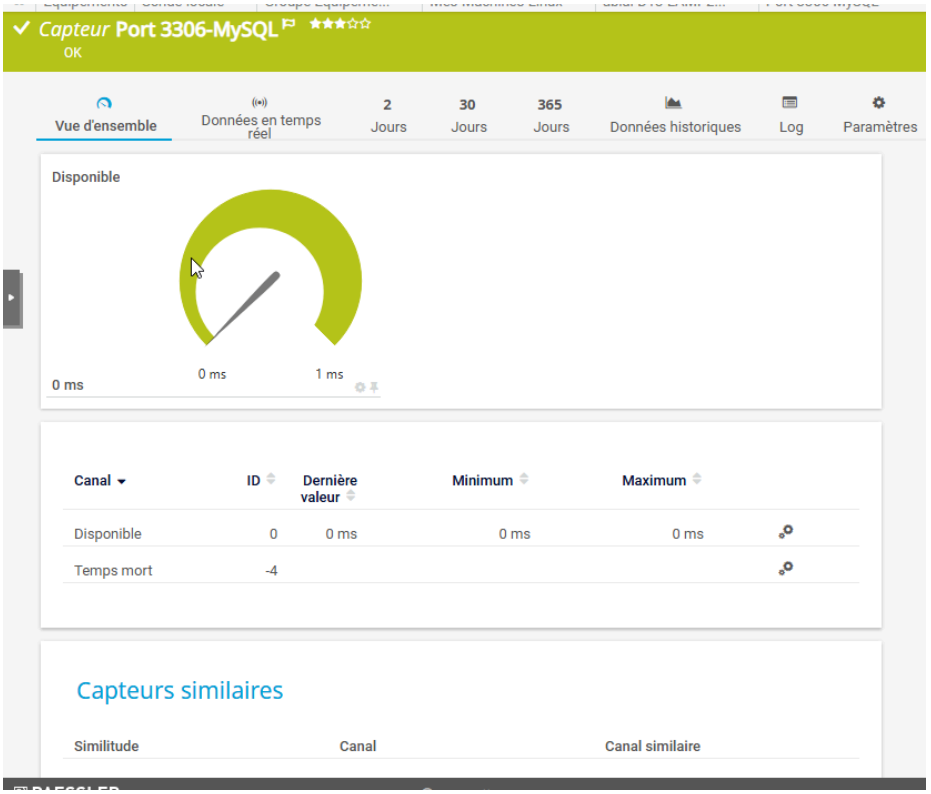
```
root@ablai-D13-LAMP:/var/www/html/siteab# cd
root@ablai-D13-LAMP:~# systemctl stop apache2
root@ablai-D13-LAMP:~#
```

Je me connecte en tant que `root` sur mon serveur Linux et j'exécute la commande `systemctl stop apache2`.
Objectif : En coupant manuellement le service Apache, je simule une panne du serveur web pour forcer le capteur à passer en état d'alerte.

Suite à l'arrêt du service, les capteurs HTTP v2, HTTPS v2 et mon capteur personnalisé siteab passent instantanément au rouge.
Cela prouve que ma configuration est fonctionnelle. Le système de supervision détecte bien l'indisponibilité de la page web et m'alerte de la panne sur ma machine Linux.

1.4 Supervision du service MySQL (Port 3306)

Pour garantir la disponibilité de la base de données MySQL au sein de la pile LAMP, j'ai configuré un capteur spécifique pour le port TCP 3306.



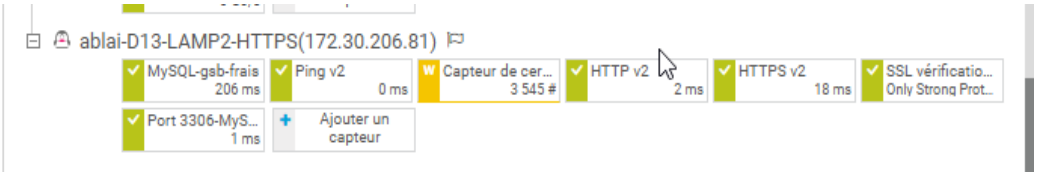
Le capteur vérifie que le service MySQL est bien "en écoute" sur le réseau.
Le tableau de bord affiche un statut "Disponible" (en vert) avec un temps de réponse très faible (0 ms), ce qui confirme que la communication avec la base de données fonctionne avec succès.

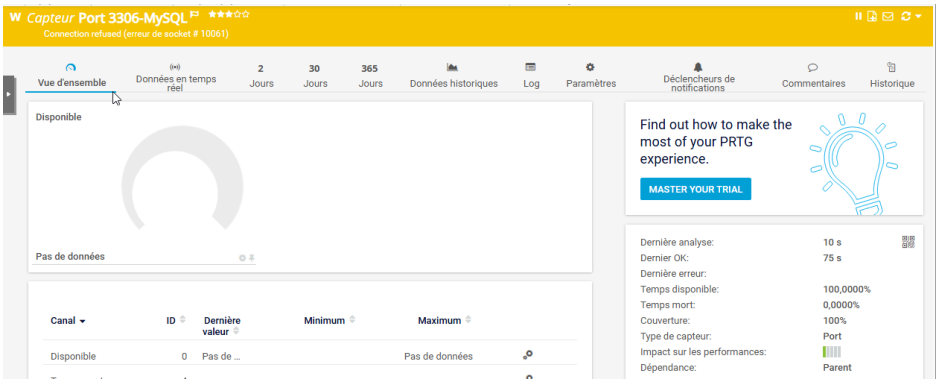
1.5 Procédure de test par arrêt du service

Tout comme pour le service Apache, je procède à un test de rupture pour valider l'alerte.

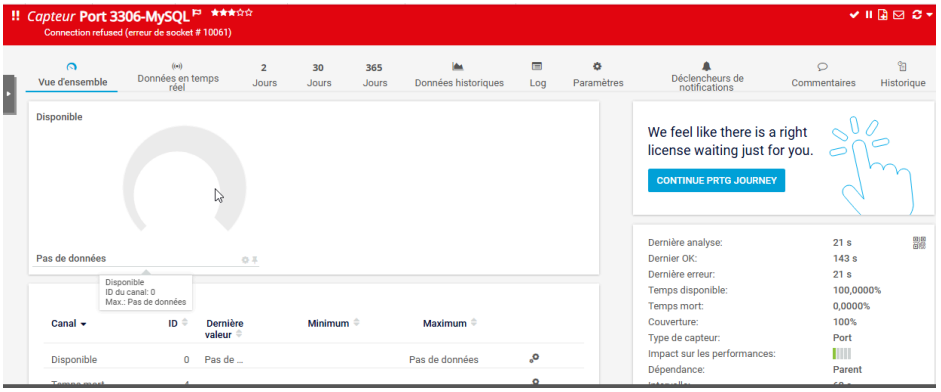
```
root@ablai-D13-HTTPS:~# service mysql stop
root@ablai-D13-HTTPS:~#
```

J'utilise la commande `service mysql stop` sur mon serveur. Cette action coupe immédiatement le processus de base de données.





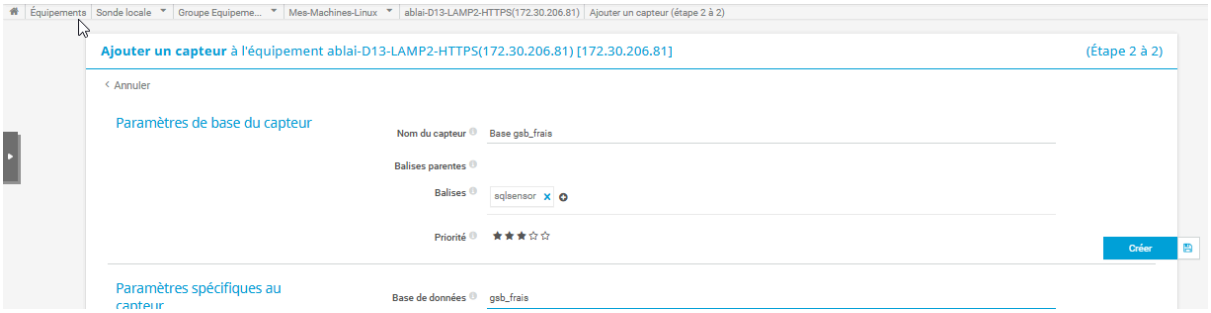
Le capteur met un peu de temps avant d'induire le signalement en rouge, le temps de faire une multitude de tests.



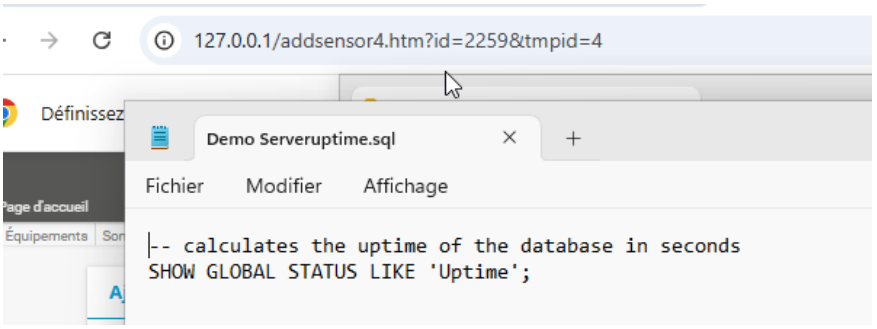
La capture montre que PRTG génère alors une erreur précise : "Connection refused (erreur de socket # 10061)". Cela indique que le port 3306 ne répond plus aux requêtes.

2. Capteur de Base de Données "Gsb_Frais"

Après avoir surveillé le port réseau (3306), je configure ici un capteur qui interroge directement le contenu et l'état de la base de données MariaDB.



Le capteur est nommé `Base gsb_frais`.
Cible spécifique : Dans les paramètres, je renseigne le nom de la base de données cible : `gsb_frais`. Cela permet de s'assurer non seulement que le serveur SQL tourne, mais que cette base soit bien accessible.



```
Database changed
MariaDB [gsb_frais]> SHOW GLOBAL STATUS LIKE 'Uptime';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| Uptime        | 392   |
+---------------+-------+
1 row in set (0,003 sec)

MariaDB [gsb_frais]>
```

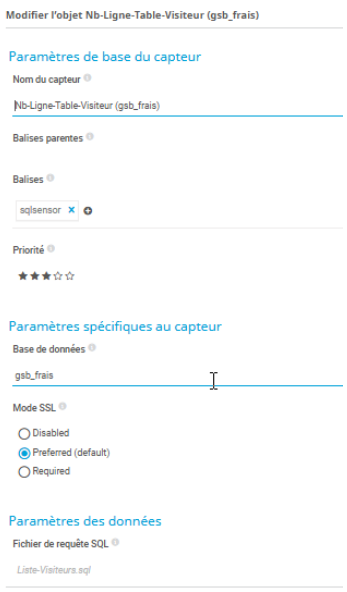
Pour mesurer la disponibilité, j'utilise une requête SQL spécifique.

La capture montre un fichier `Demo Serveruptime.sql` contenant la commande :
`SHOW GLOBAL STATUS LIKE 'Uptime';`
Cette commande retourne le temps écoulé (en secondes) depuis le dernier démarrage du serveur de base de données.
Sur la capture du terminal, on voit que la requête renvoie une valeur de **392** secondes. C'est un indicateur : si ce chiffre retombe à zéro, cela signifie que le service a redémarré.

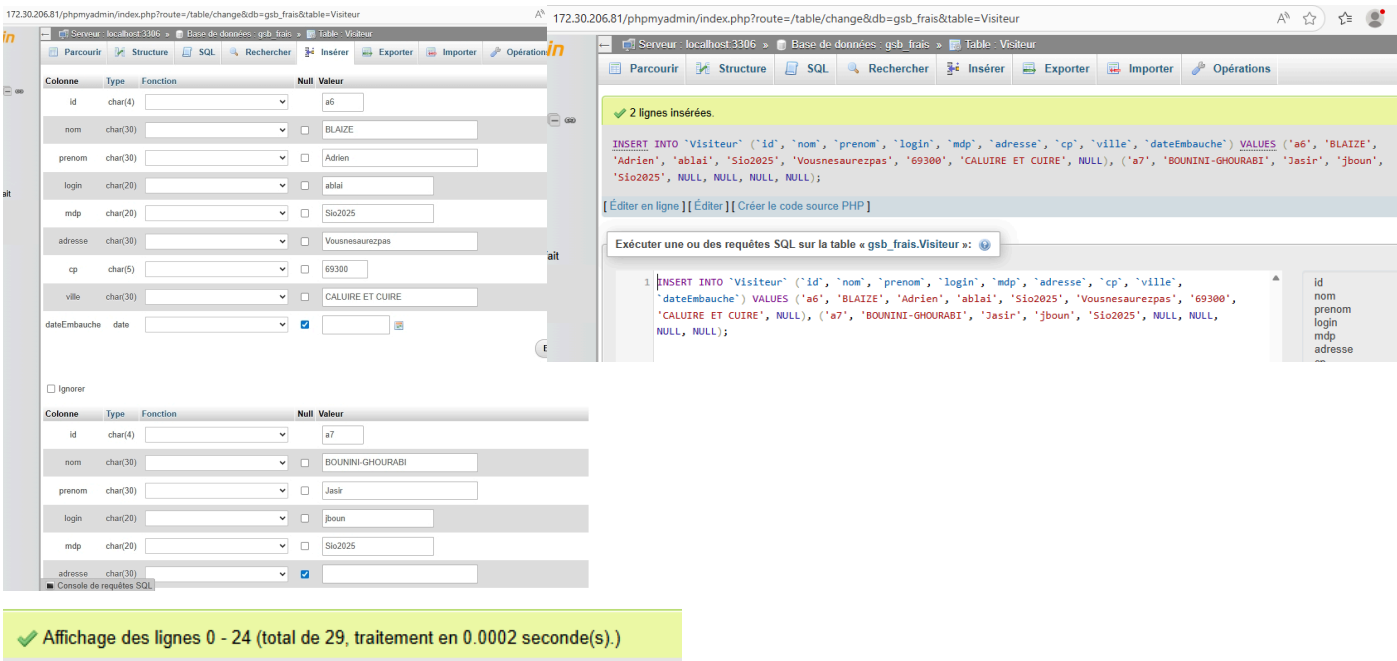
À gauche, le capteur apparaît avec un point d'interrogation gris (?), indiquant qu'il est en cours de déploiement ou d'acquisition de données.
À droite, le capteur passe au vert avec une valeur de **203 ms**. Ce temps correspond au délai d'exécution de la requête SQL. Le capteur est désormais actif et opérationnel.

3. Capteur "Liste-Visiteurs.sql"

Je crée un nouveau capteur SQL nommé `Nb-Ligne-Table-Visiteur (gsb_frais)`.



Contrairement à l'Uptime (qui surveille le temps), ce capteur va compter combien de visiteurs sont enregistrés dans la base.
J'utilise un script spécifique nommé `Liste-Visiteurs.sql`. Ce script contient probablement une commande de type `SELECT COUNT(*) FROM Visiteur;`.

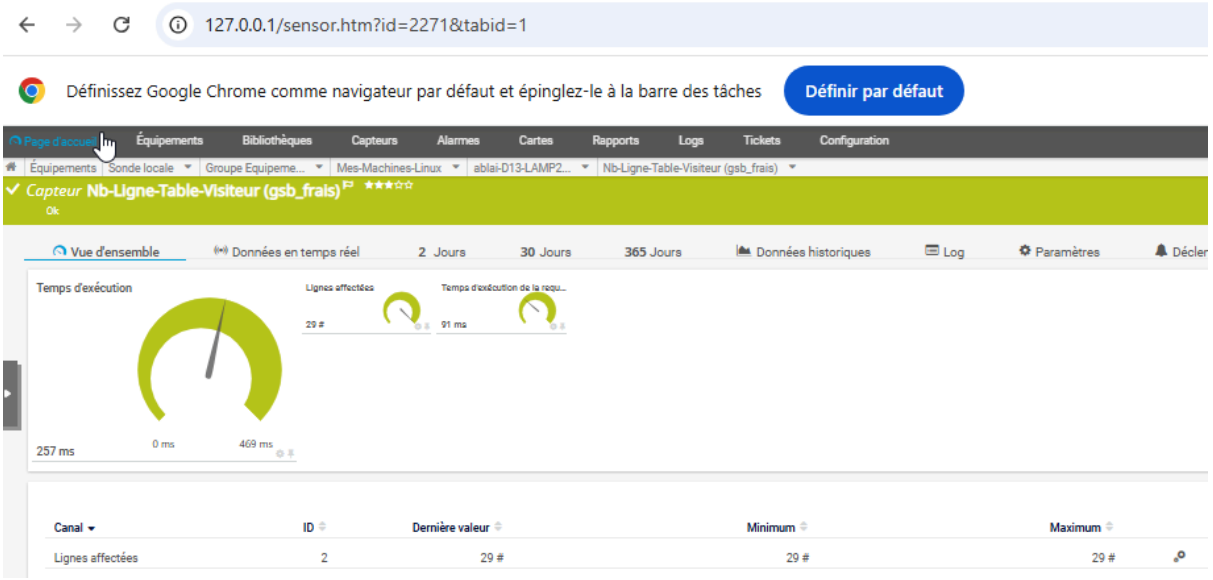


Dans l'interface phpMyAdmin, j'ajoute deux nouveaux enregistrements dans la table `Visiteur` (identifiants `a6` et `a7` pour BLAIZE et BOUNINI-GHOURABI).

Le message "2 lignes insérées" apparaît, et la table affiche désormais un total de 29 lignes.

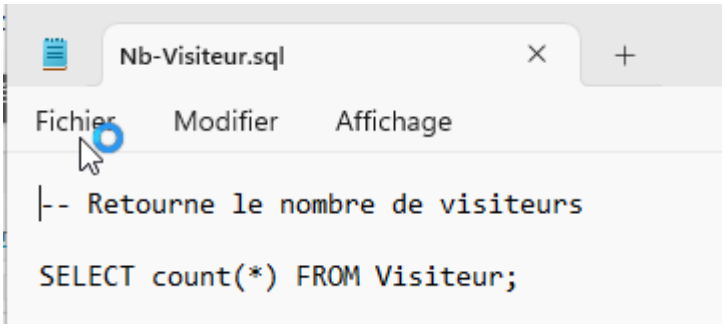
	id	nom	prenom	login	mdp	adresse	cp	ville	dateEmbauche
Éditer Copier Supprimer	a131	Villechalane	Louis	lvillachane	jux7g	8 rue des Charmes	46000	Cahors	2005-12-21
Éditer Copier Supprimer	a17	Andre	David	dandre	oppg5	1 rue Petit	46200	Lalbenque	1998-11-23
Éditer Copier Supprimer	a55	Bedos	Christian	cbedos	gmhxd	1 rue Peranud	46250	Montcuq	1995-01-12
Éditer Copier Supprimer	a6	BLAIZE	Adrien	ablai	Sio2025	Vousnesaurezpas	69300	CALUIRE ET CUIRE	*NULL*
Éditer Copier Supprimer	a7	BOUNINI-GHOURABI	Jasir	jboun	Sio2025	*NULL*	*NULL*	*NULL*	*NULL*

Je constate que les users Adrien et Jasir ont été ajoutés avec succès.



Je constate que mon capteur affiche désormais la valeur 29 #.
Cela me confirme que mon script SQL personnalisé fonctionne parfaitement. Je ne me contente plus de savoir si mon serveur est "allumé" ou "éteint" : je surveille désormais l'intégrité de mes données. Si un problème survenait et que des données venaient à disparaître, je serais immédiatement alerté par une chute de cette valeur sur mes graphiques.

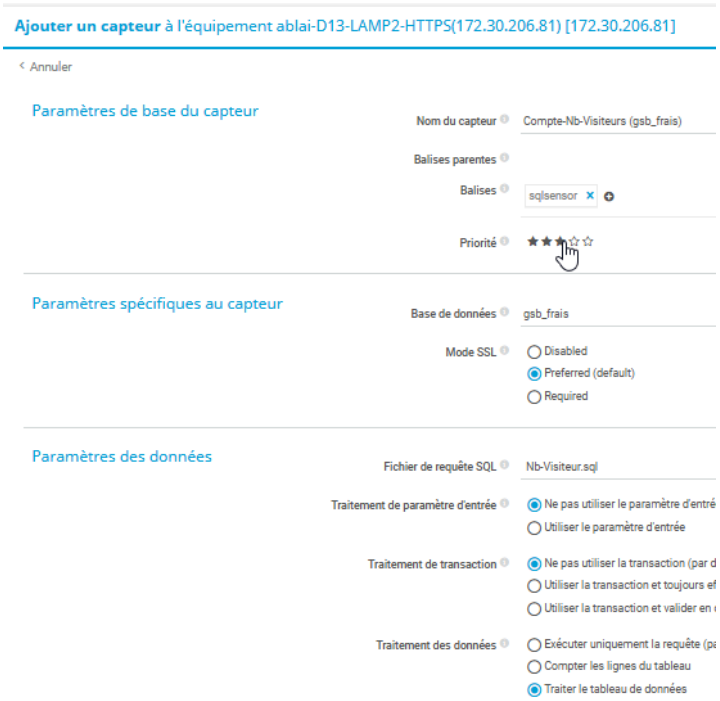
4. Capteur "Nb-Visiteur.sql"



Pour superviser le contenu de ma base, j'ai commencé par rédiger un script SQL simple que j'ai nommé `Nb-Visiteur.sql`.

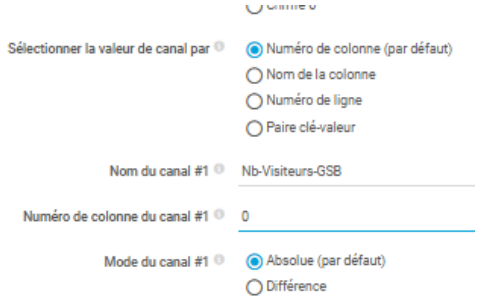
J'ai utilisé l'instruction `SELECT count(*) FROM Visiteur;`.

Cette requête me permet de retourner un nombre entier correspondant au total des enregistrements dans ma table. C'est ce chiffre que je souhaite suivre dans mon outil de supervision.



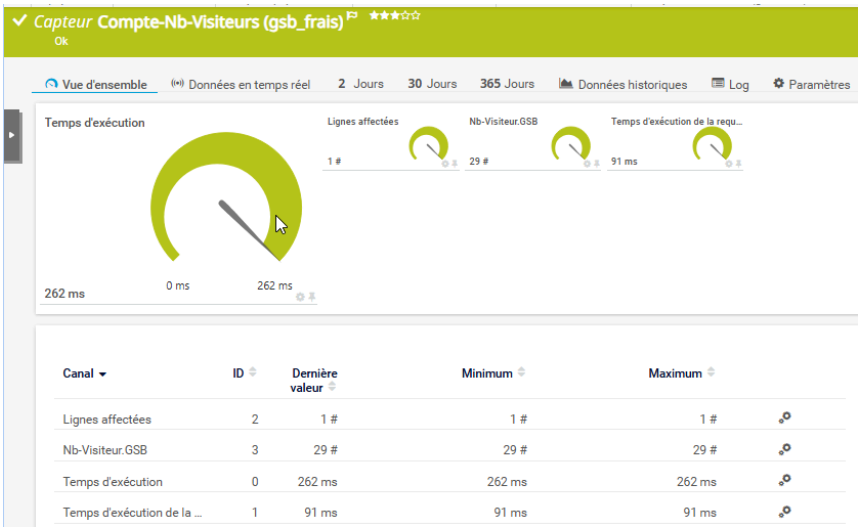
J'ai ensuite intégré ce script dans un nouveau capteur sur mon équipement `ablai-D13-LAMP2-HTTPS`.

J'ai nommé mon capteur `Compte-Nb-Visiteurs (gsb_frais)`. J'ai spécifié la base de données cible `gsb_frais` et j'ai lié mon fichier de requête `Nb-Visiteur.sql`.



Pour que PRTG interprète correctement le résultat, j'ai configuré le Canal #1 avec le nom `Nb-Visiteurs-GSB`. J'ai réglé le mode sur "Absolue" car je veux afficher le nombre total exact de visiteurs à chaque relevé.

Une fois les paramètres enregistrés, j'observe la mise en service du capteur en bas de l'écran. Le capteur apparaît d'abord avec une icône grise (point d'interrogation) indiquant qu'il est en phase d'initialisation, puis il passe rapidement en mode scan (icône orange avec le "W" de Waiting). Cela me confirme que ma requête est en cours d'exécution sur le serveur MariaDB.



Je constate que la valeur affichée est de 29 #. Cela correspond exactement au nombre de lignes que j'avais vérifié précédemment dans ma table `Visiteur`. Mon capteur est donc parfaitement fiable.
Je note que le temps d'exécution global est de 262 ms.

Le temps d'exécution de la requête SQL seule est de 91 ms.
Cette différence m'indique le temps de traitement interne de PRTG et la latence réseau. Ces chiffres sont excellents et montrent que ma pile LAMP est performante.

4. Supervision du SWITCH VLAN 206

4.1 Configuration de mon capteur d'état système

Pour garantir la haute disponibilité de mon infrastructure, je ne surveille pas seulement mes serveurs, mais aussi mes équipements d'interconnexion. J'ai ici configuré un capteur spécifique pour mon commutateur Cisco.

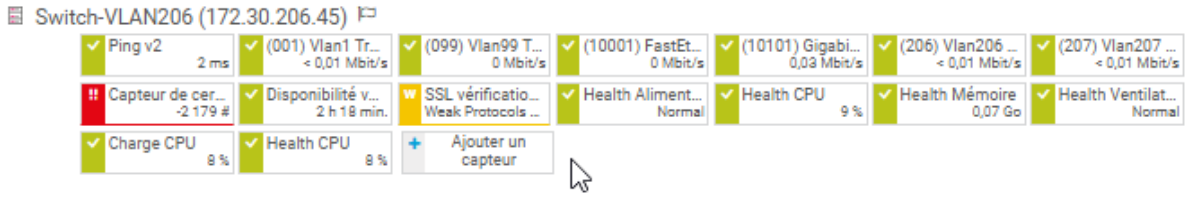
Il s'agit du `Switch-VLAN206`, identifié par l'adresse IP `172.30.206.45`.
Ma sélection de mesures : Dans les paramètres du capteur (utilisant le protocole SNMP), j'ai choisi de surveiller l'intégralité des composants critiques du châssis :

Pour prévenir toute saturation du plan de contrôle.

Pour anticiper une panne matérielle physique qui pourrait paralyser le réseau du VLAN 206.

4.2 Validation de l'état de santé du switch

Une fois le capteur ajouté, je vérifie le tableau de bord de mon équipement pour m'assurer que tout est opérationnel.

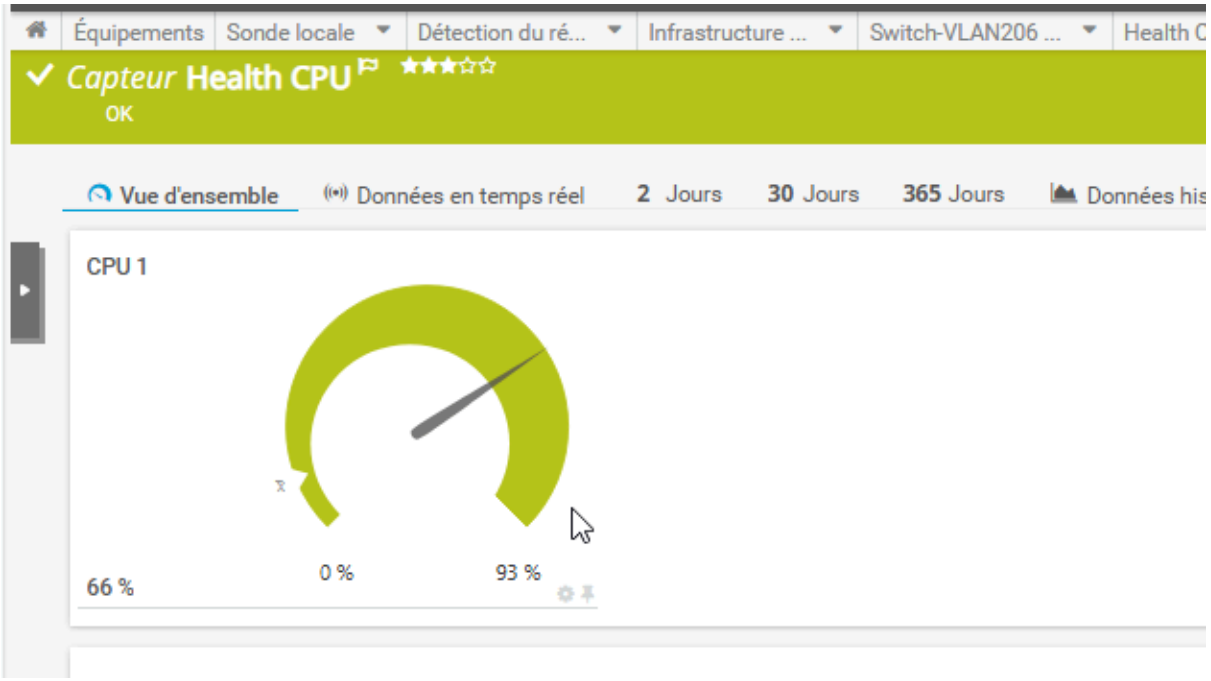


La majorité de mes capteurs sont au vert, ce qui me confirme que le switch fonctionne dans des conditions nominales.
Mes indicateurs de performance :

Charge CPU : Je note une charge très faible de 8% ou 9%, ce qui est idéal.

Santé matérielle : Les capteurs `Health Aliment...`, `Health CPU` et `Health Ventilateur` affichent tous l'état "Normal".

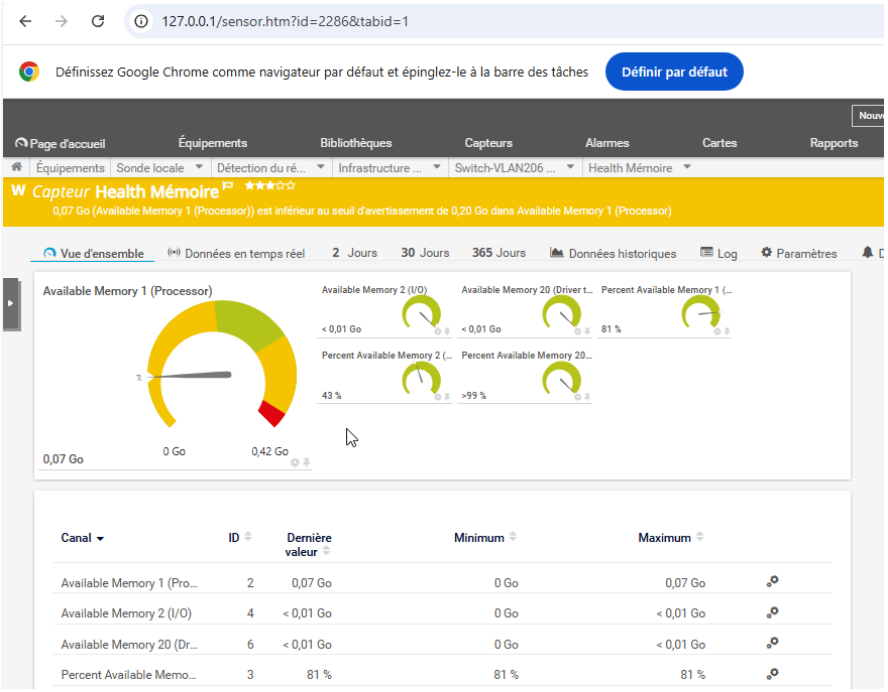
Je vois que la consommation est stable à 0,07 Go.



Suite à une attaque sur le Switch, incluant des dizaines de milliers de pings afin d'exploiter le CPU à minima pour ce test, on peut affirmer que le capteur reçoit avec succès les informations le concernant.

4.3 Surveillance de la mémoire du Switch

J'ai mis en place une surveillance approfondie de la mémoire de mon commutateur pour éviter tout ralentissement réseau.



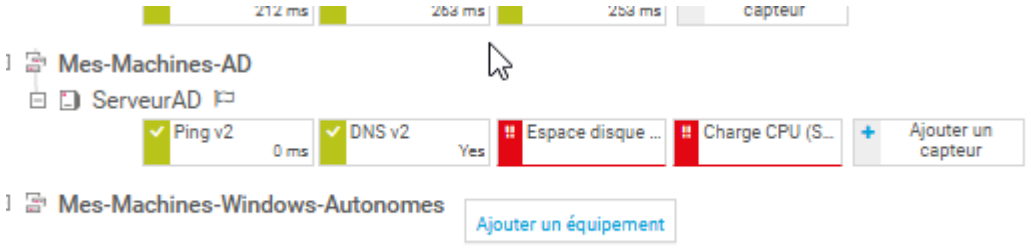
Le capteur affiche un état d'avertissement (orange).

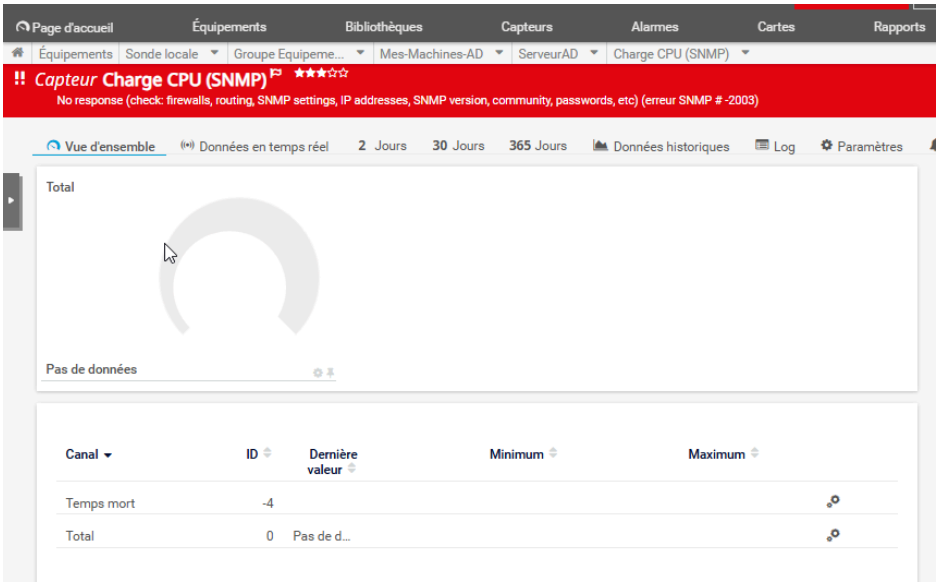
Le message m'indique que la mémoire disponible (0,07 Go) est inférieure au seuil d'avertissement que j'avais fixé à 0,20 Go.

Bien que le système fonctionne encore avec 81% de mémoire disponible pour le processeur, ce seuil m'alerte sur une possible fuite de mémoire ou une configuration trop lourde. Je devrai ajuster mes seuils ou surveiller l'évolution de cette consommation pour éviter un plantage de l'équipement.

5. Supervision de mon Serveur Active Directory

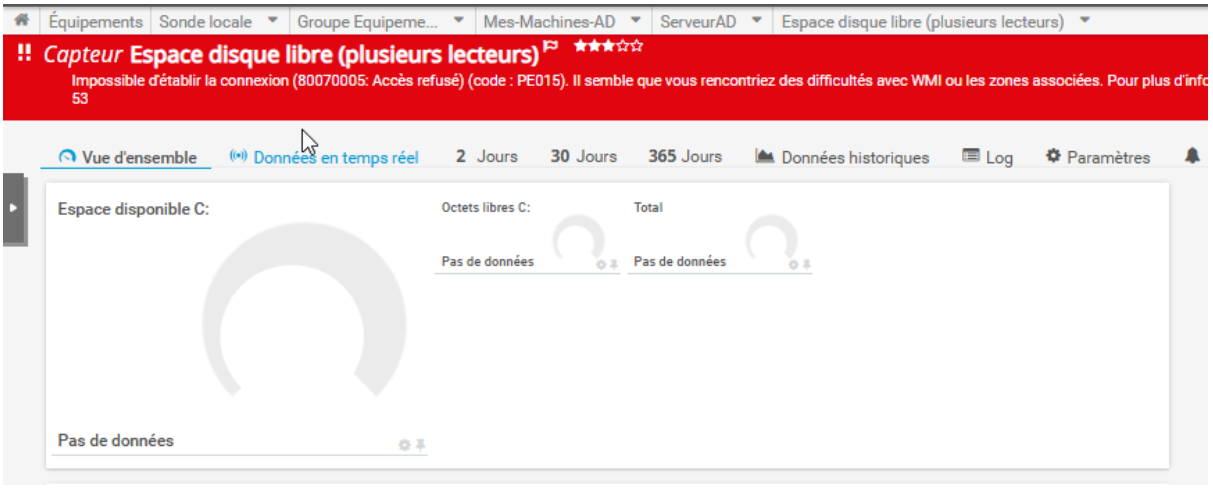
Je passe ensuite à la supervision de mon serveur de domaine (`ServeurAD`) situé dans le groupe `Mes-Machines-AD`.





Je vois que les services de base comme le Ping v2 et le DNS v2 sont au vert. Cela me confirme que mon serveur est joignable et qu'il remplit bien son rôle de résolution de noms.
Le capteur est en erreur. Cela signifie que le service SNMP sur mon serveur Windows n'est pas correctement configuré pour répondre à PRTG.

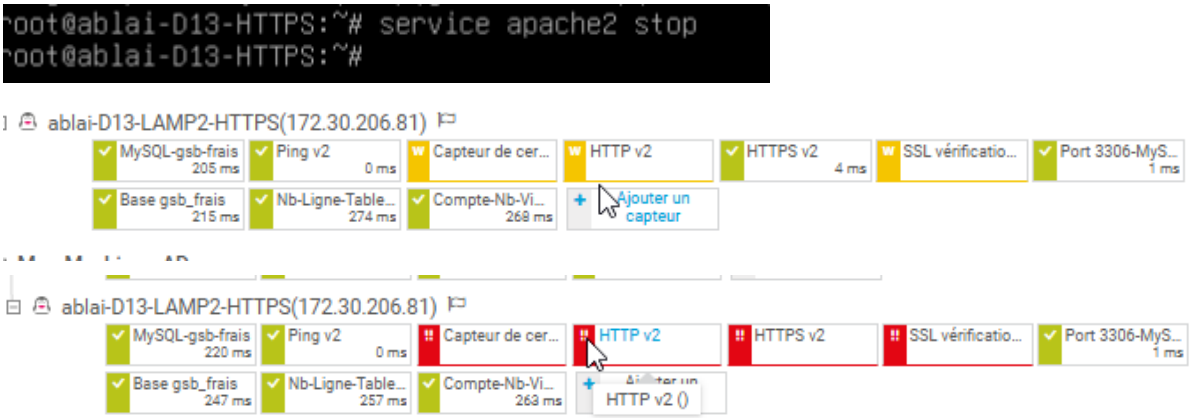
6. Supervision de mon disque dur



Ce capteur est également au rouge. Des complications à initié le capteur sont survenues.

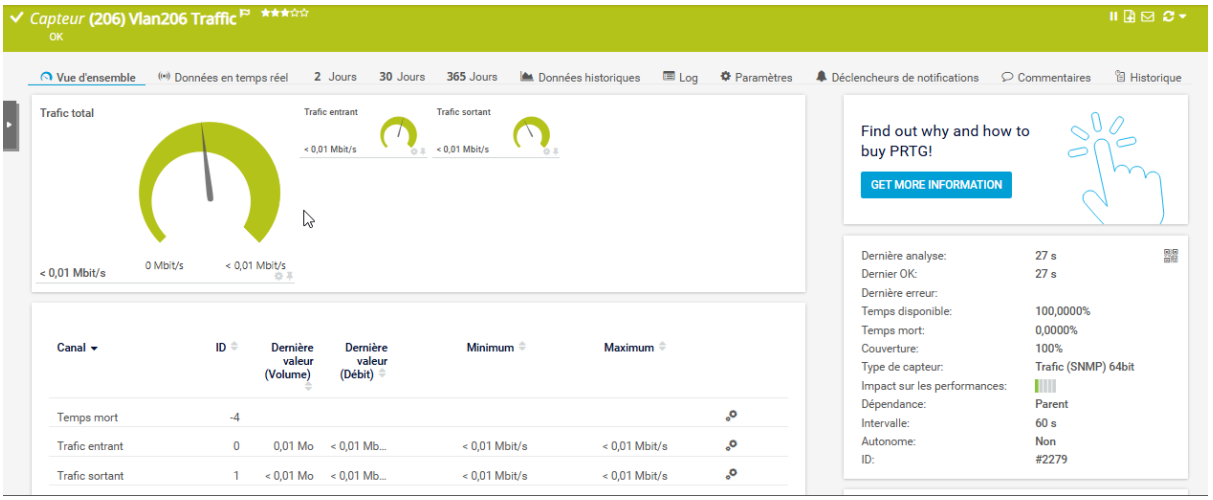
6. Capteurs créés automatiquement

6.1 Capteur HTTP & co

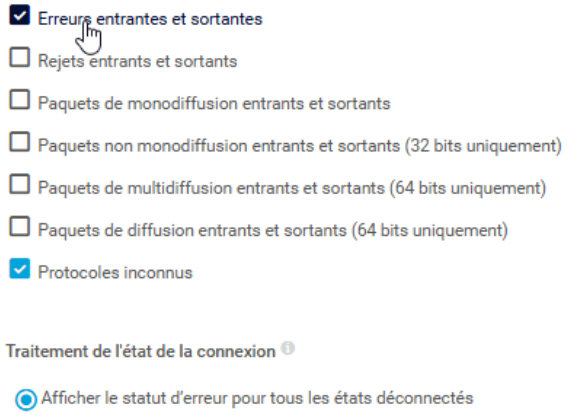


À l'instant précis où j'ai coupé le service Apache, PRTG a enregistré l'alerte. PRTG indique clairement la raison du passage en rouge (par exemple, un délai d'attente dépassé ou une connexion refusée).

7. Capteur Switch-VLAN206

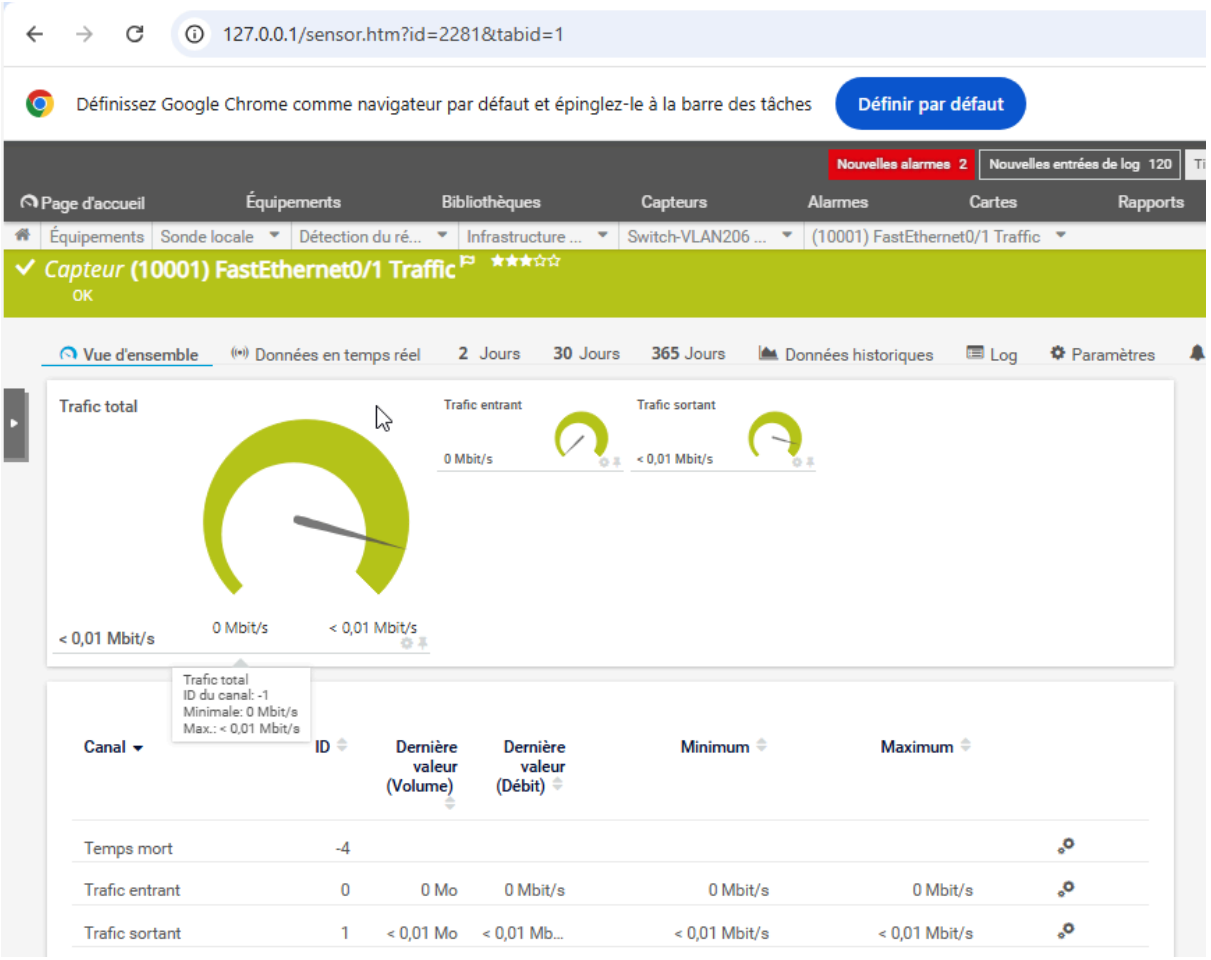


Le voyant est vert (OK). Le temps de disponibilité (Uptime) est à 100%, ce qui signifie qu'il n'y a pas eu de coupure de communication entre PRTG et le switch. Actuellement, le trafic est extrêmement faible (< 0,01 Mbit/s). Les jauges circulaires sont presque à zéro, ce qui prouve que ce VLAN est très peu sollicité au moment de la capture mais que le capteur fonctionne.

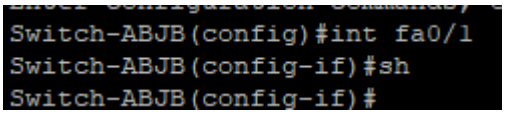


Erreurs entrantes et sortantes : Cela permet de voir si des paquets sont corrompus ou perdus à cause d'un problème physique (câble défectueux, interférences) ou d'une mauvaise configuration de port.
Protocoles inconnus : Cela compte les paquets que le switch ne sait pas identifier. Si ce chiffre grimpe, cela peut indiquer un équipement mal configuré ou un trafic réseau inhabituel.

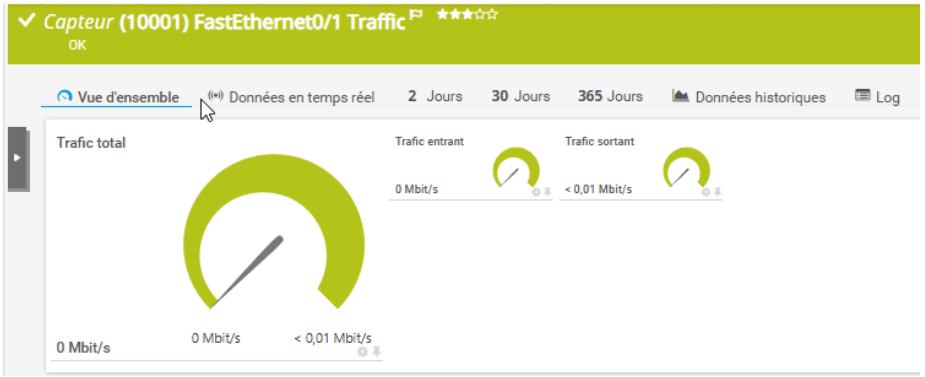
8. Le Capteur PRTG (FastEthernet 0/1)



Contrairement au VLAN 206 qui est une interface logique, ici je surveille le câble physique branché sur le port fa0/1.
Les jauges indiquent 0 Mbit/s. Le port est "Up" (vert), mais il n'y a absolument aucun trafic qui circule à l'instant T.
On voit un historique de volume inférieur à 0,01 Mo, ce qui confirme que l'équipement au bout de ce câble ne communique pas (ou est éteint/en veille).

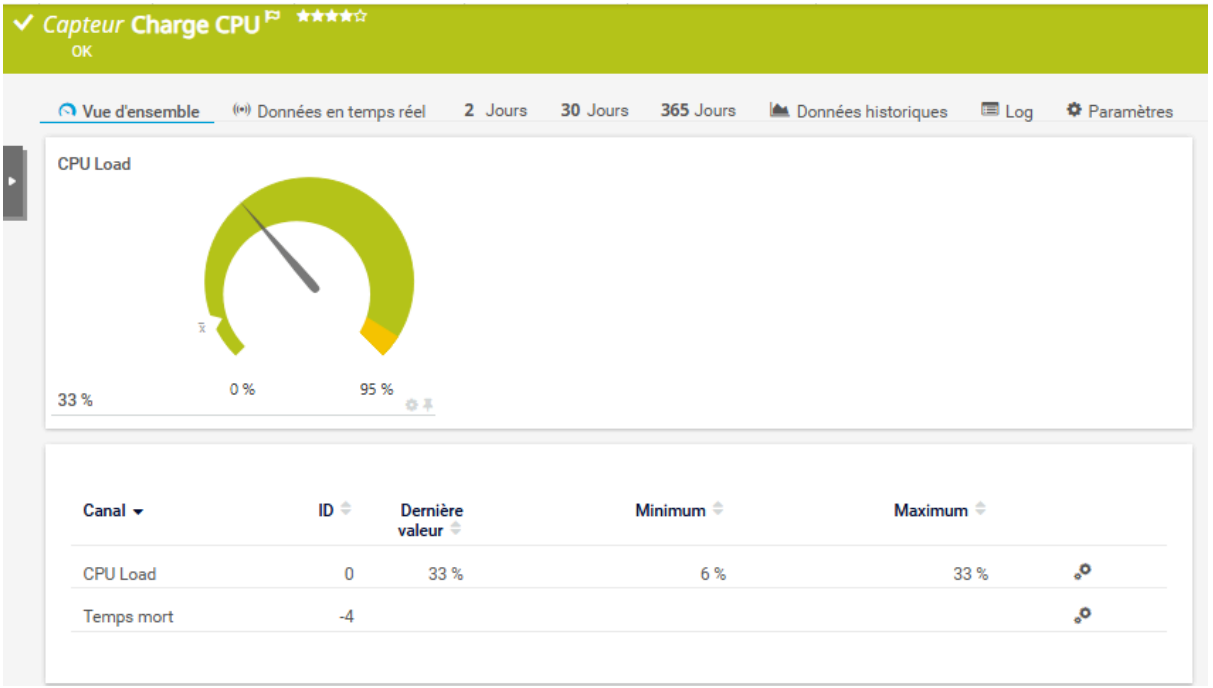


J’ai shutdown l’interface pour prouver l’efficacité du capteur :



Une minute après avoir shutdown l'interface, le capteur ne détecte plus aucun signal et cela prouve son efficacité.

9. Le Capteur "Charge CPU"

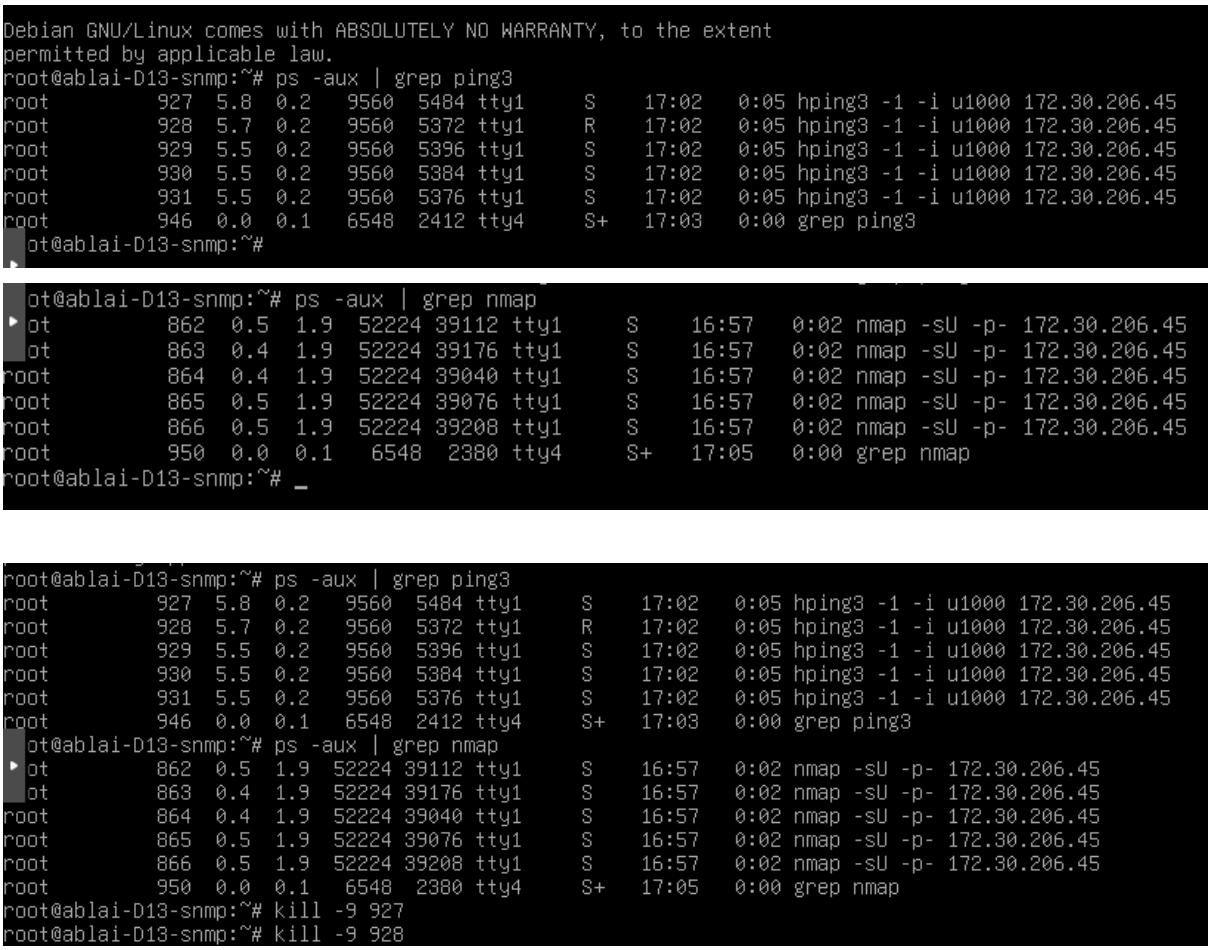


L'aiguille est montée à 33 %.
Le processeur de l'équipement surveillé (probablement le switch ou le serveur cible) commence à travailler. Ce n'est pas encore une saturation critique (le rouge est à 95 %), mais on voit nettement l'impact de tes commandes de stress par rapport au repos total des captures précédentes.

9.1 Génération de trafic intensif (Stress Test)

```
root@ablai-D13-snmp:~# hping3 -1 -i u1000 172.30.206.45 &
len=46 ip=172.30.206.45 ttl=255 id=53291 icmp_seq=1348 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=63685 icmp_seq=1349 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=3664 icmp_seq=1350 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=35605 icmp_seq=1351 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=58534 icmp_seq=1352 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=19057 icmp_seq=1353 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=42779 icmp_seq=1354 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=10560 icmp_seq=1355 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=32095 icmp_seq=1356 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=61816 icmp_seq=1357 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=2716 icmp_seq=1358 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=17294 icmp_seq=1359 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=1065 icmp_seq=1360 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=3633 icmp_seq=1361 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=52260 icmp_seq=1362 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=17708 icmp_seq=1363 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=38453 icmp_seq=1364 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=62978 icmp_seq=1365 rtt=0.0 ms
len=46 ip=172.30.206.45 ttl=255 id=12847 icmp_seq=1366 rtt=0.0 ms
^C
```

Avec hping3, j'envoie chaque paquet avec une intervalle de 1000 microsecondes entre chacun d'entre eux.

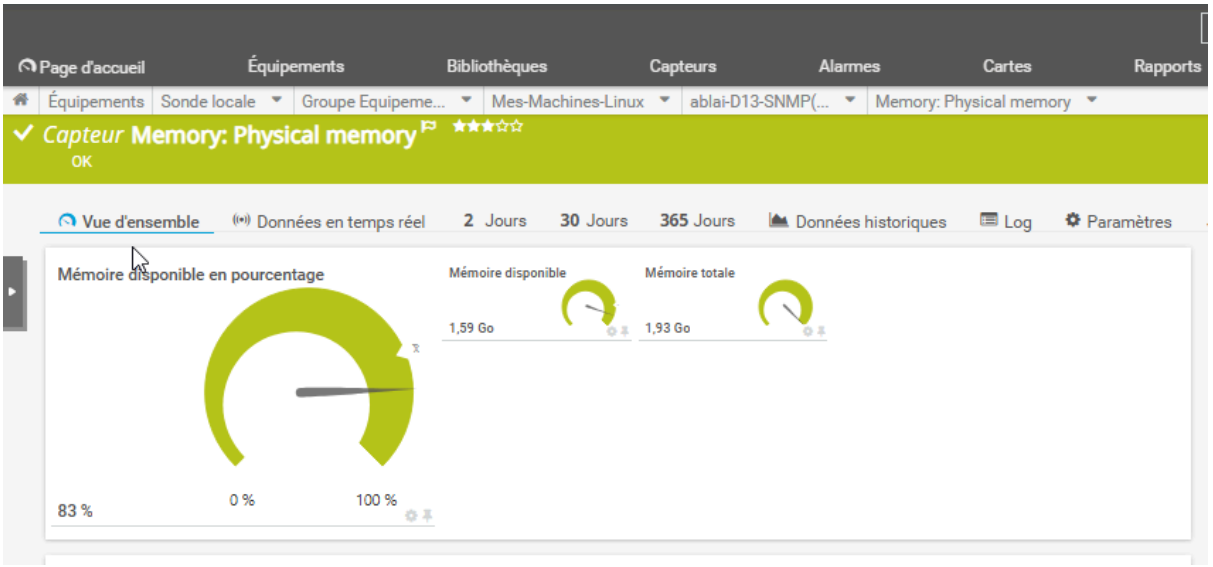


J'utilise la commande `ps -aux | grep [nom]` pour lister les processus actifs liés à mes tests de charge (`hping3` et `nmap`).
On identifie clairement 5 instances de hping3 (PID 927 à 931) consommant chacune environ 5,5% à 5,8% de CPU.

On identifie également 5 instances de nmap (PID 862 à 866) effectuant un scan UDP intensif (`-sU -p-`).

J'exécute la commande `kill -9 [PID]` pour forcer l'arrêt immédiat des processus. Dans l'exemple, les PID 927 et 928 sont supprimés.

10. Surveillance de la RAM (Physical Memory)



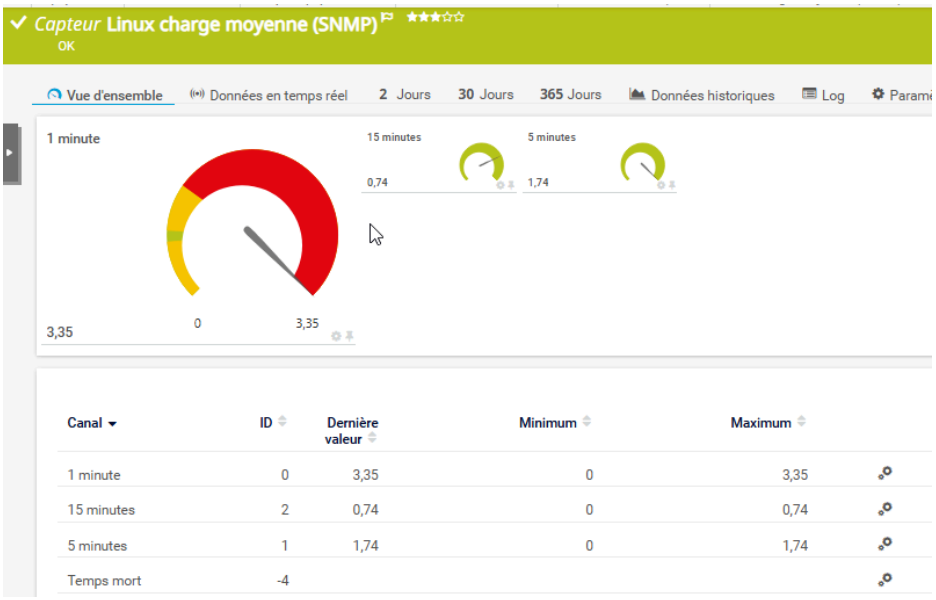
Le capteur PRTG indique une mémoire disponible de 83 % (soit environ 1,39 Go libres sur 1,63 Go au total).
Le système dispose de suffisamment de ressources de stockage temporaire.
L'utilisation de la mémoire vive reste stable malgré les tests réseau effectués précédemment. Il reste bien plus haut que d'habitude.

11. Le capteur "Linux charge moyenne (SNMP)"

11.1 Utilisation de l'outil "Stress" (Linux)

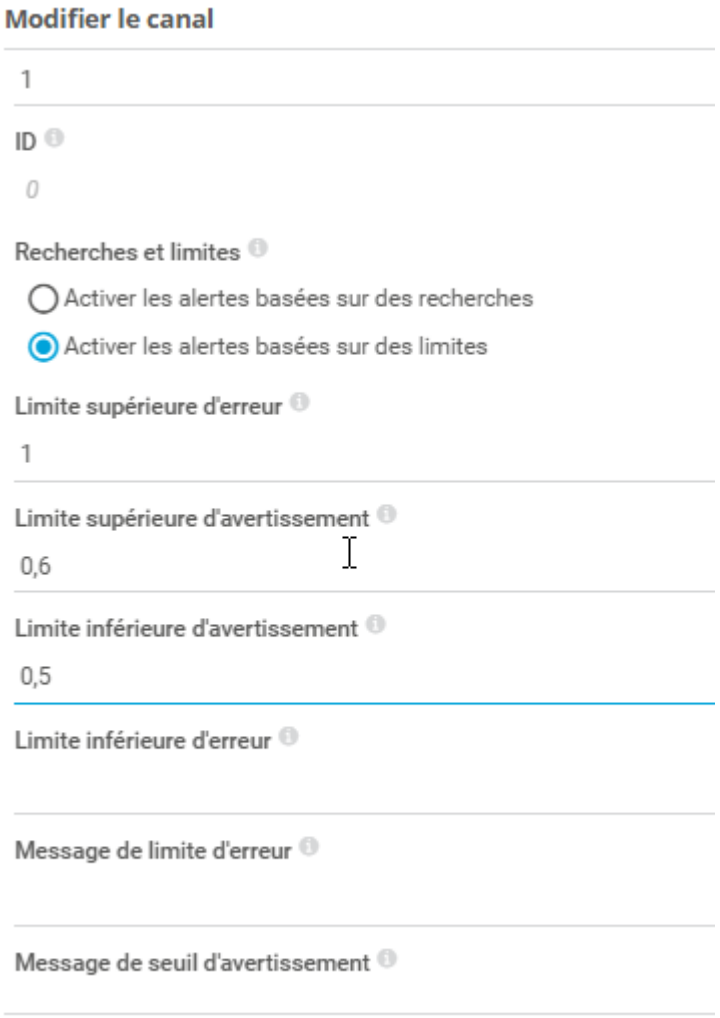
```
root@ablai-D13-snmp:~# stress -c 4 --timeout 100
stress: info: [1076] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
stress: info: [1076] successful run completed in 100s
root@ablai-D13-snmp:~#
```

L'argument `-c 4` demande à l'outil de générer une charge maximale sur 4 cœurs CPU simultanément.
Le `--timeout 100` programme l'arrêt automatique du test après 100 secondes.



Le capteur "Linux charge moyenne (SNMP)" est passé dans le rouge. La jauge "1 minute" atteint une valeur de 3,35.
Cette valeur indique une saturation. PRTG détecte immédiatement l'anomalie de performance grâce aux compteurs SNMP. La différence entre les jauges "1 min", "5 min" et "15 min" permet de visualiser l'évolution de la surcharge dans le temps.

11.2 Configuration des seuils d'alerte (Limites)

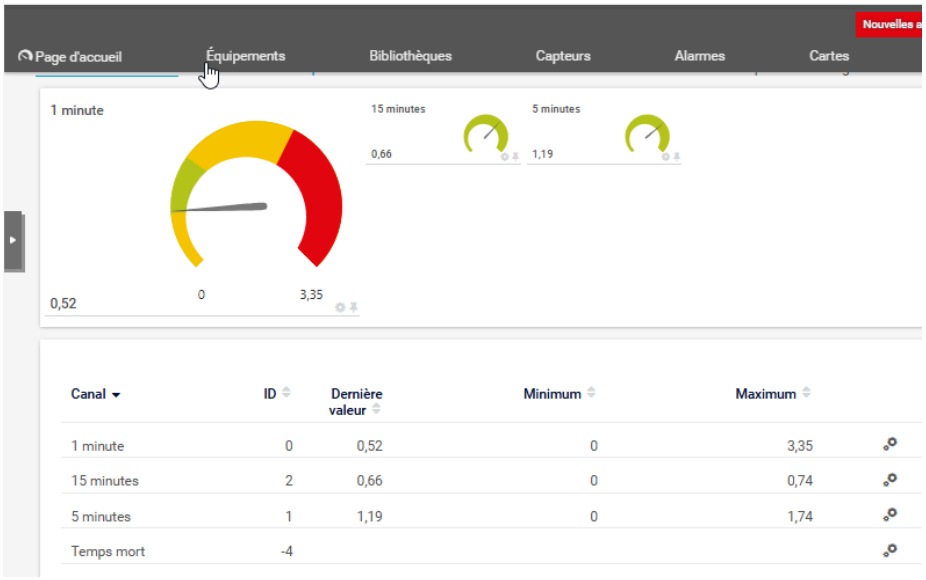


Dans les paramètres du canal "1 minute" de la charge moyenne, j'ai activé les alertes basées sur des limites.
Détail des seuils :

Limite supérieure d'erreur : 1. Si la charge dépasse 1 pendant 1 minute, le capteur passe au rouge (État "Down").

Limite supérieure d'avertissement : 0,6. Entre 0,6 et 1, le capteur passera au jaune.

Le bandeau de PRTG devient rouge vif. Le message indique : *"3,35 (1 minute) est supérieur au seuil d'erreur de 1 dans 1 minute"*.
Suite au test de charge effectué avec l'outil `stress`, la valeur réelle (3,35) a largement dépassé la limite autorisée (1). L'alerte est immédiatement levée.



Dans le cadre d'ajouts d'alertes sur le capteur, j'ai rajouté des alertes en fonction :

Si l'aiguille de la jauge "1 minute" redescend vers 0,52 (zone jaune). Les valeurs pour "5 minutes" (1,19) et "15 minutes" (0,66) sont encore un peu élevées mais en baisse.

La jauge "1 minute" réagit le plus vite, tandis que les moyennes à plus long terme mettent plus de temps à s'apurer.

Le capteur repassera au vert dès que la charge "1 minute" passera sous le seuil d'avertissement de 0,6 que j'ai configuré précédemment.