LIGNIERE
Timothée
BTS2 SIO
Date de création
12/10/23
Date de maj
15/04/24

Grafana

Ce tutoriel a été fait sur une machine ayant Prometheus d'installé

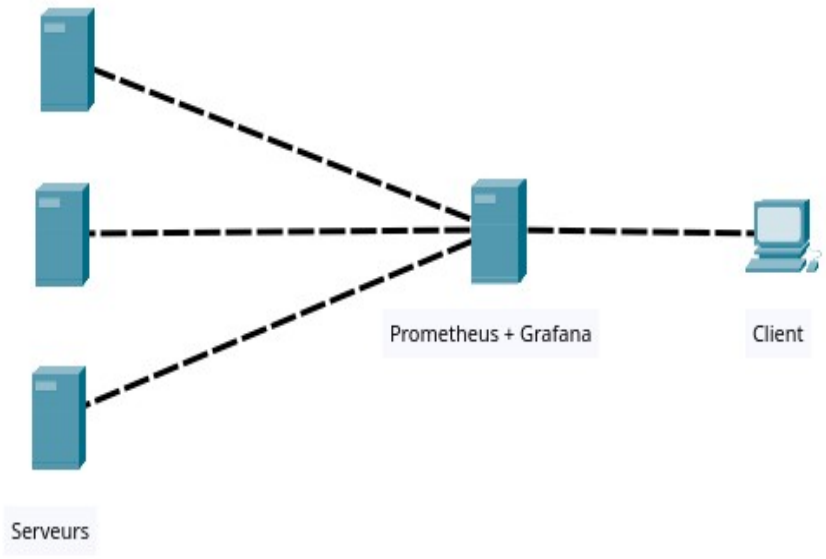
Sommaire

1 - Problématique

Lors de l'utilisation de Prometheus, il peut être fastidieux de se remémorer de chaque commande pour vérifier le bon fonctionnement de toutes les machines et des services.

Grafana résout cela en proposant des graphiques modifiables et modulables pour avoir toutes les informations déjà traitées au préalable visuellement.

2 - Schéma logique de la solution



3 - Notice d’installation

3.1 - Prérequis

Pour commencer, nous allons installer des paquets qui permettront d’assurer l’installation :

```
sudo apt-get install -y apt-transport-https software-properties-common wget ufw
```

Nous allons rajouter le miroir de grafana pour télécharger le logiciel :

```
sudo mkdir -p /etc/apt/keyrings/

wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null

echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

Ensuite, vous devriez pouvoir installer le paquet de Grafana :

```
apt install grafana-enterprise
```

Désormais, démarré le serveur Grafana et activez-le pour les futurs redémarrages de votre machine

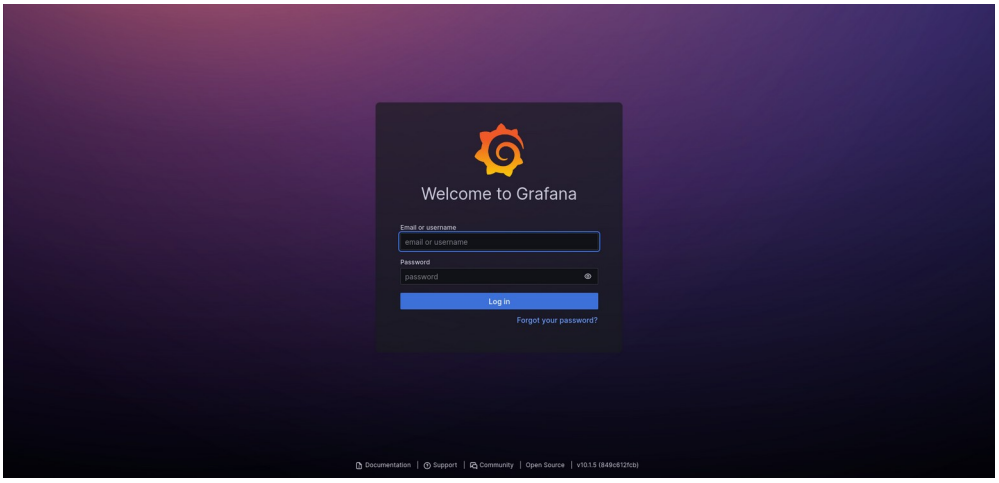
```
Systemctl enable grafana-server
Systemctl start grafana-server
```

Si le serveur fonctionne bien, vous devriez pouvoir, vous connectez au port 3000 de votre machine. Cependant, il se peut que le pare-feu ne vous laisse pas passer ; rajoutez donc une règle pour laisser le port 3000 ouvert.

```
ufw allow 3000
```

3.2 – Mise en relation avec Prometheus

Pour cette partie, vous devez vous connecter au port 3000 de votre



machine. Ce qui devrait vous redirigez vers cette page :

Les identifiants lors de la première connexion sont :

```
Identifiant : admin
Mot de passe : admin
```

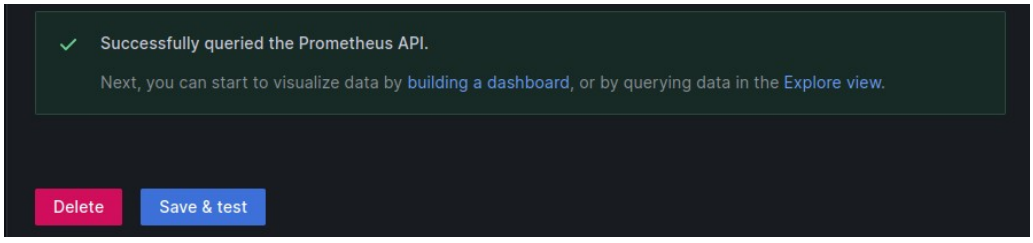
Ensuite, Grafana vous proposera de changer le mot de passe ; faites à votre guise.

Après cela vous aurez la page d'accueil de Grafana. Nous allons donc rajouter une source de données avec l'onglet « Add your first data source »

Choisissez « Prometheus » (en théorie, il devrait être en haut de la page), Grafana vous ouvrira ensuite une page pour le relié aux informations de Prometheus. Dans mon cas Prometheus se trouve sur la même machine, je donne donc à Grafana l'adresse suivante pour « Prometheus server URL » :

```
http://127.0.0.1:9090
```

Sauvegarder lors que vous êtes satisfait de vos paramètres. Grafana devrait vous indiquez si il a bien réussi a contacter le service Prometheus



Nous allons pouvoir passer à la mise en place des Dashboards (Tableaux de bord) dans la partie Notice d'utilisation de ce tutoriel

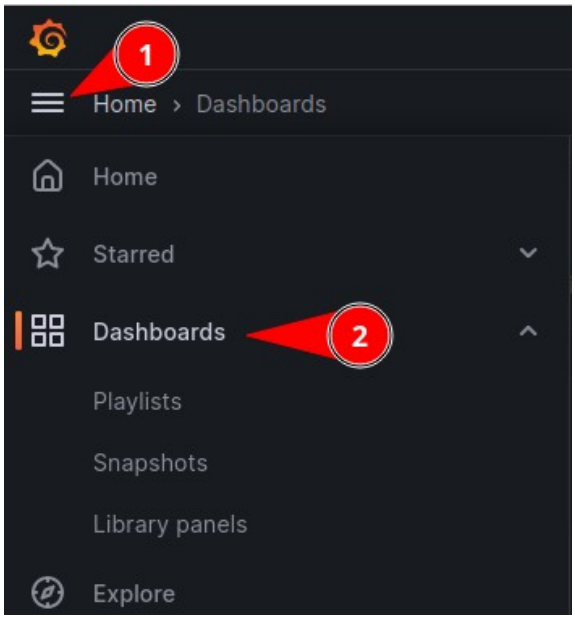
4 - Notice d'utilisation

4.1 – Création de Dashboard

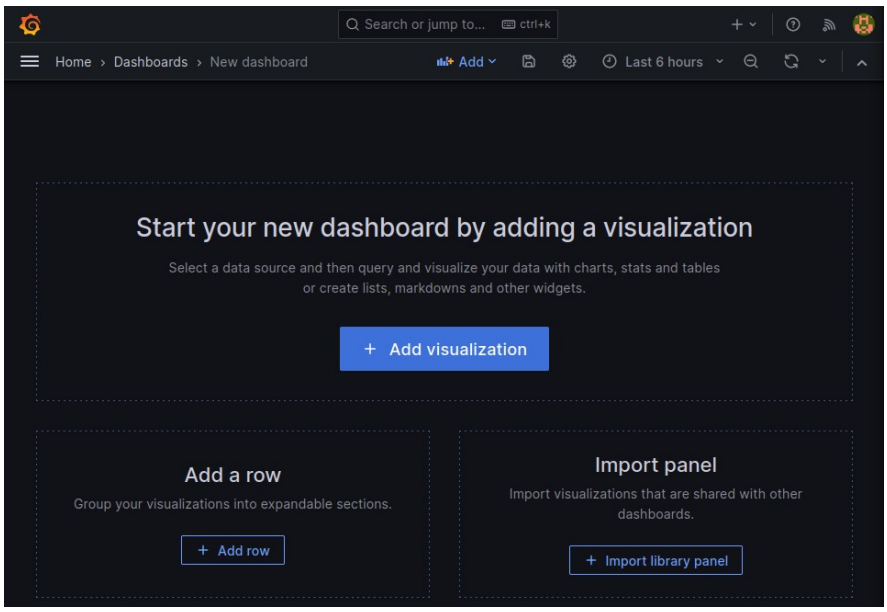
Si vous souhaitez importer un dashboard préfait avec les graphiques présentés dans cette doc, se référer à https://github.com/TLigniere/Personal_configuration_files.git

4.1.1 – Première fois / up {job= »node »}

Toujours sur le site de Grafana, cliquez sur les 3 barres horizontales en haut a droite puis sur Dashboards :

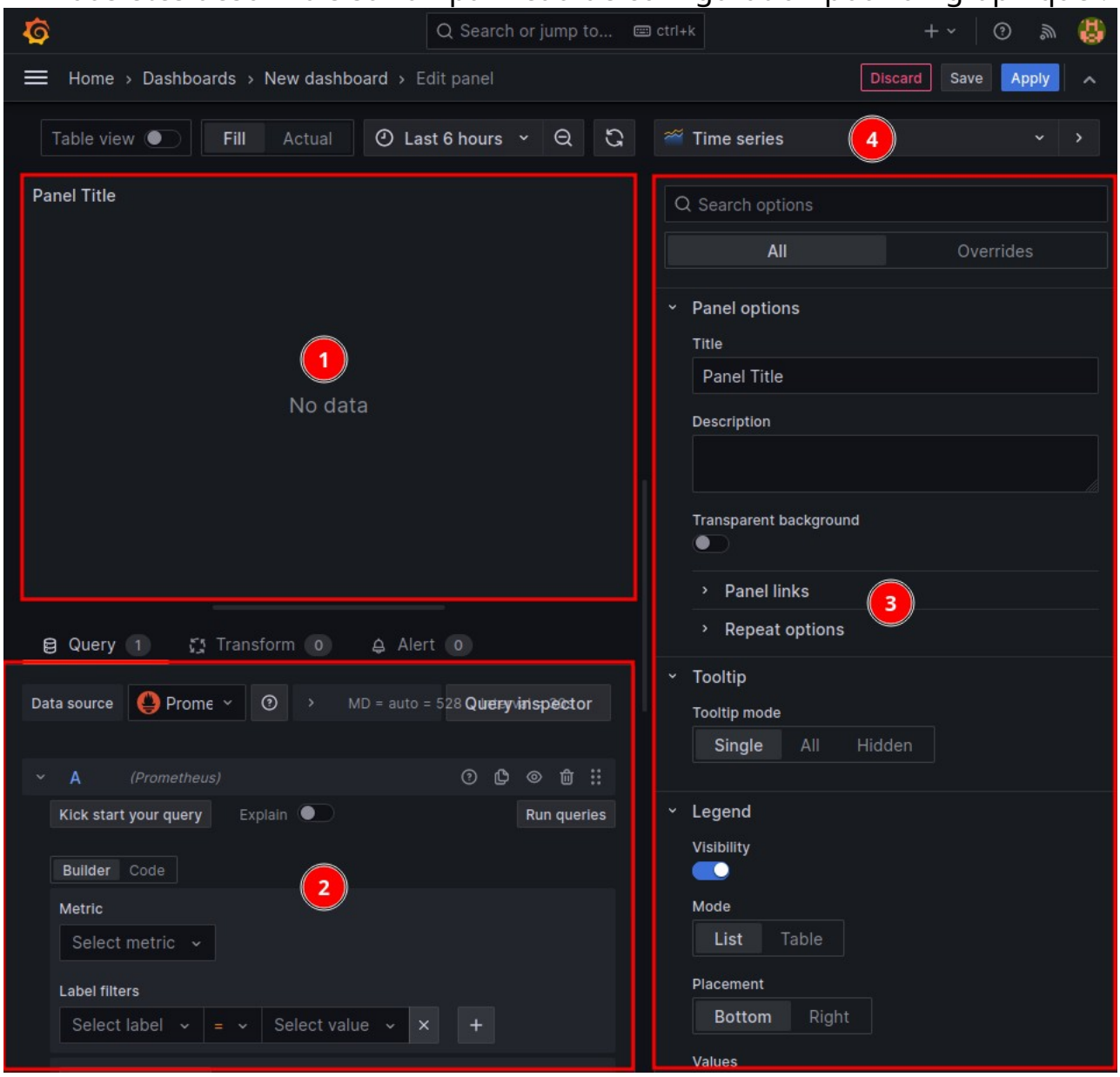


Ensuite, cliquez sur « New » puis « New dashboard », vous devriez avoir ceci désormais :



Cliquez sur « Add visualization », puis sur une source de données (dans notre cas, Prometheus)

Vous êtes désormais sur un panneau de configuration pour un graphique :



1 : Panneau d'affichage = Il permet de voir le graphique que nous sommes en train de créer afin de le modifier selon nos besoins.

2 : Paramétrages de métriques = Permet de rentrer des commandes afin d'indiquer à Grafana quels infos il doit faire apparaître.

3 : Paramètres d'affichage = Permet de modifier des paramètres tel que la couleur des données, le style du graphique, les horaires des données, etc.

4 : Paramètres d'affichage généraux : Permet de changer le type de graphique (Jauges, Statistiques, Graphique en barre, Courbes,etc)

Nous allons donc faire plusieurs graphiques pour les besoins les plus généraux :

- Le taux d'utilisation du processeur
- Le taux d'utilisation de la mémoire vive
- Le taux de mémoire libre sur le disque dur
- Vérifier que chaque node_exporter de Prometheus soit bien fonctionnels

Pour rajouter une valeur dans les paramètres de métriques vous avez 2 options de mise en place :

Builder : Permet de rajouter des valeurs grâce à des blocs préfabriqués

Code : Permet de directement donner une ligne de commande PromQL.

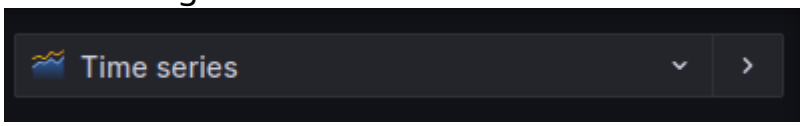
Pour commencer nous allons simplement rajouter un graphique pour vérifier le fonctionnement des node_exporters.

Mettez vous en mode code et rajouter ceci :

```
up{job="node"}
```

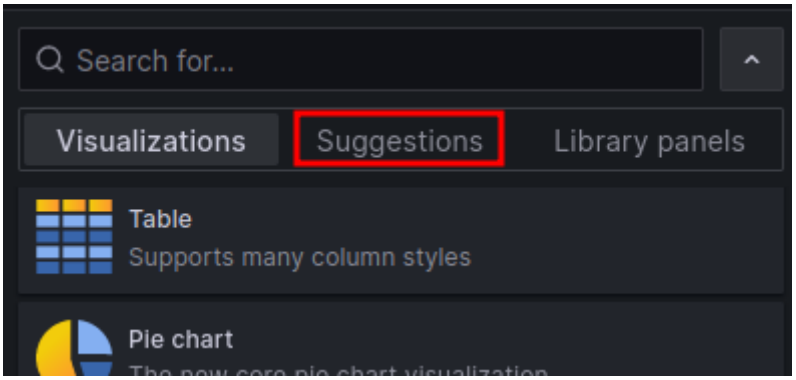
Il n'est pas impossible que la tâche (job= »node ») ait un autre nom de votre côté selon la manière que vous avez configuré le prometheus.yml.

Puis, étant donné que « up » ne peut que retourner une valeur égale à 0 ou 1, on va changer d'affichage :



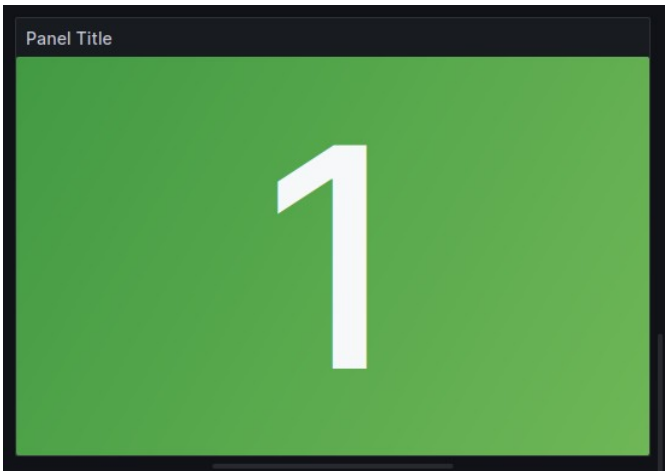
En cliquant sur la flèche diriger vers le bas, une liste avec différentes propositions apparaît, vous pouvez ainsi chercher ce qui vous convient le plus.

Cependant, vous pouvez vous simplifier la tâche avec le bouton « Suggestions » :



Avec « Suggestions », Grafana essayera de deviner l'affichage qui vous convient le plus grâce aux valeurs à sa disposition.

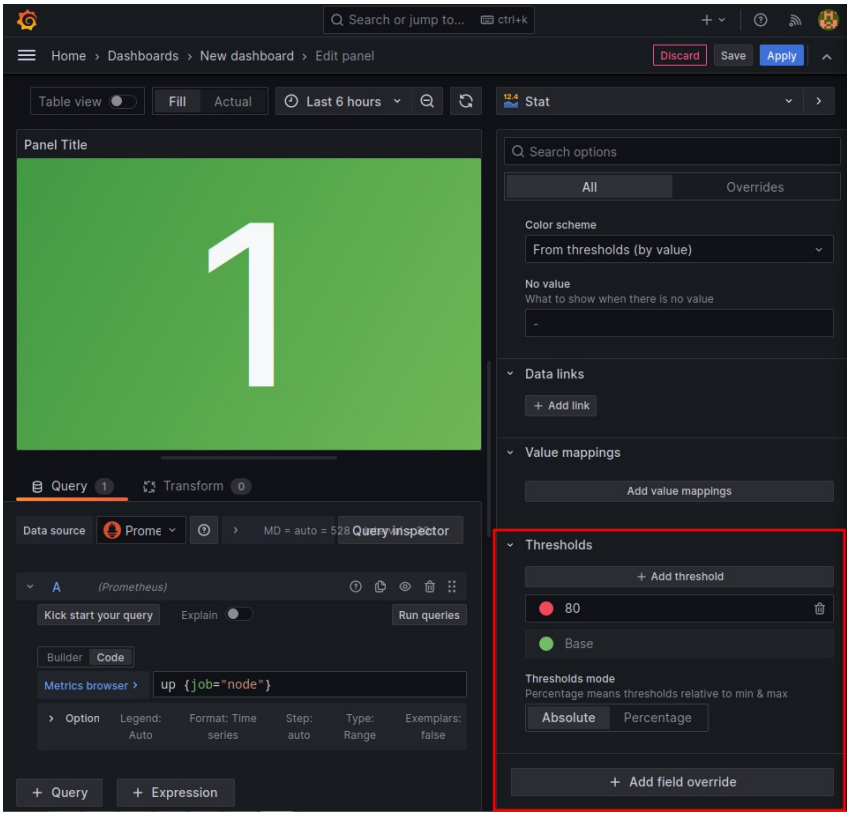
Personnellement, je favorise donc l'affichage suivant pour vérifier le fonctionnement des nodes :



Par ailleurs, si vous avez plusieurs machines, Grafana s'adaptera de lui même pour chaque case de chaque node.

Nous avons encore 2 paramètres à modifier pour être optimal en termes de présentation.

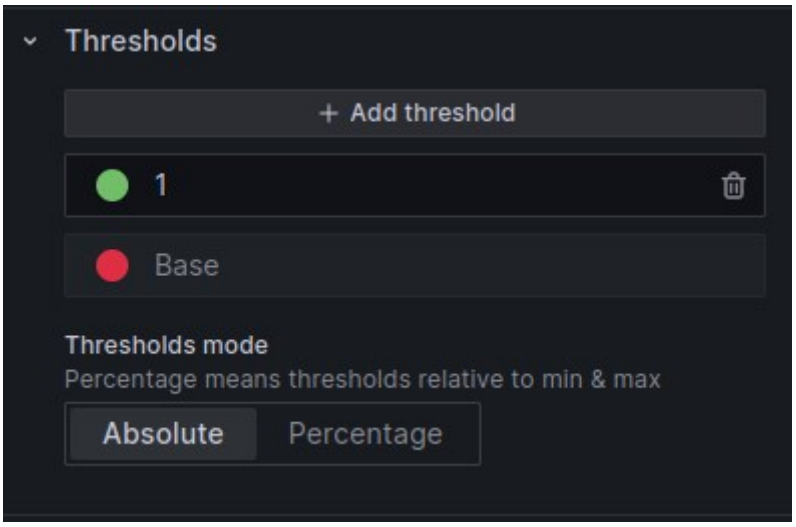
Pour le premier paramètre, descendez tout en bas du menu à droite de votre écran jusqu'à « Tresholds » :



Ce paramètre définit la couleur de notre case selon la valeur renvoyée à Grafana.

Initialement, le tableau devient donc rouge quand la valeur dépasse 80 qu'importe l'unité et la source de la valeur (vous pouvez changez de de valeur absolue à pourcentage mais cela nous est pas utile pour l'instant) . Nous allons donc changer ça pour s'adapter à « up »

Changez la couleur de la valeur de base en rouge et la couleur de 80 à vert, tout en changeant la valeur de 80 à 1 :



Ainsi quand la valeur passera à 0, (ce qui indiquera que le node est inactif) la case deviendra rouge, le rendant facilement visible en parallèle des autres valeurs que nous allons mettre dans notre tableau.

Pour le deuxième paramètre pour être optimal, nous allons adapter le paramètre des métriques pour savoir à quel valeur correspond à quelle machine. **Ce paramètre repose sur le prometheus.yml de prometheus, si vous ne l'avez pas configuré pour que chaque node ait un nom de job différents, alors cela ne servira à rien.**

Addendum 4.1.1 – Prometheus .yml

```
  # If prometheus-node-exporter is installed, grab stats about the local
  # machine by default.
  static_configs:
    - targets: ['localhost:9100']
      labels:
        hostname: "MON-server"
```

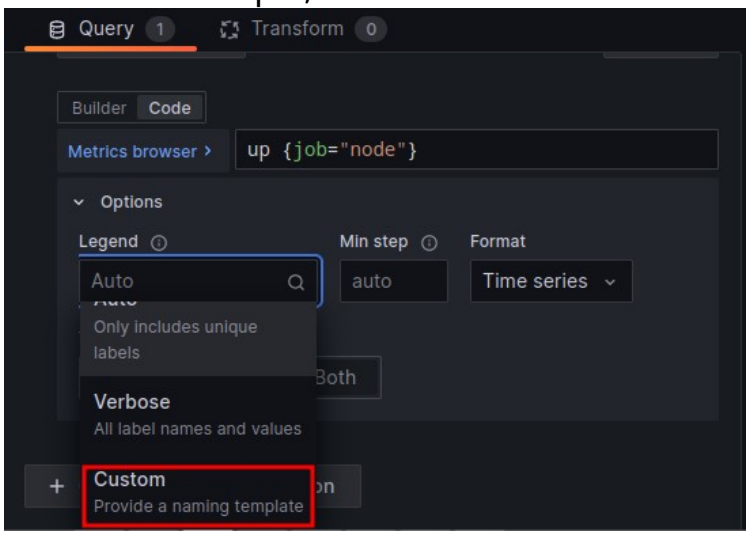
Dans le fichier /etc/prometheus/prometheus.yml de votre machine ayant Prometheus, vous devez rajouter pour chacune des targets le paramètre suivant :

```
labels:
 hostname: « <Nom de votre machine> »
```

Le nom de votre machine n'a pas besoin d'avoir de valeur particulière, prenez donc le nom qui vous convient le plus.

Fin de l'addendum

Dans les options de la métrique, choisissez le mode « Custom »

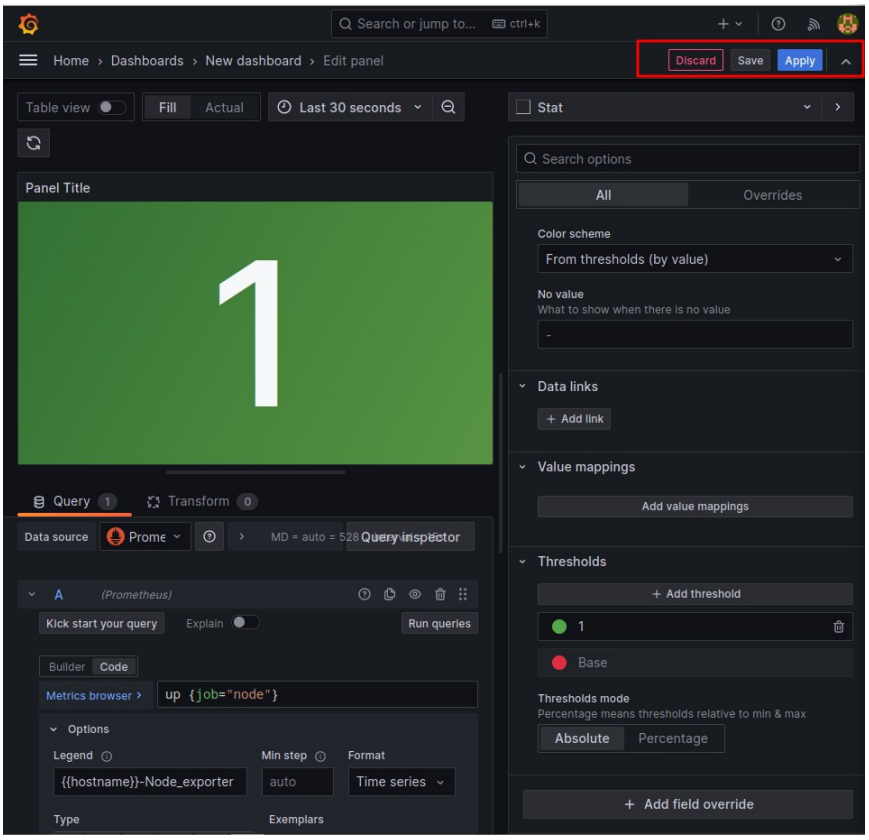


Puis mettez en paramètre la légende suivante :

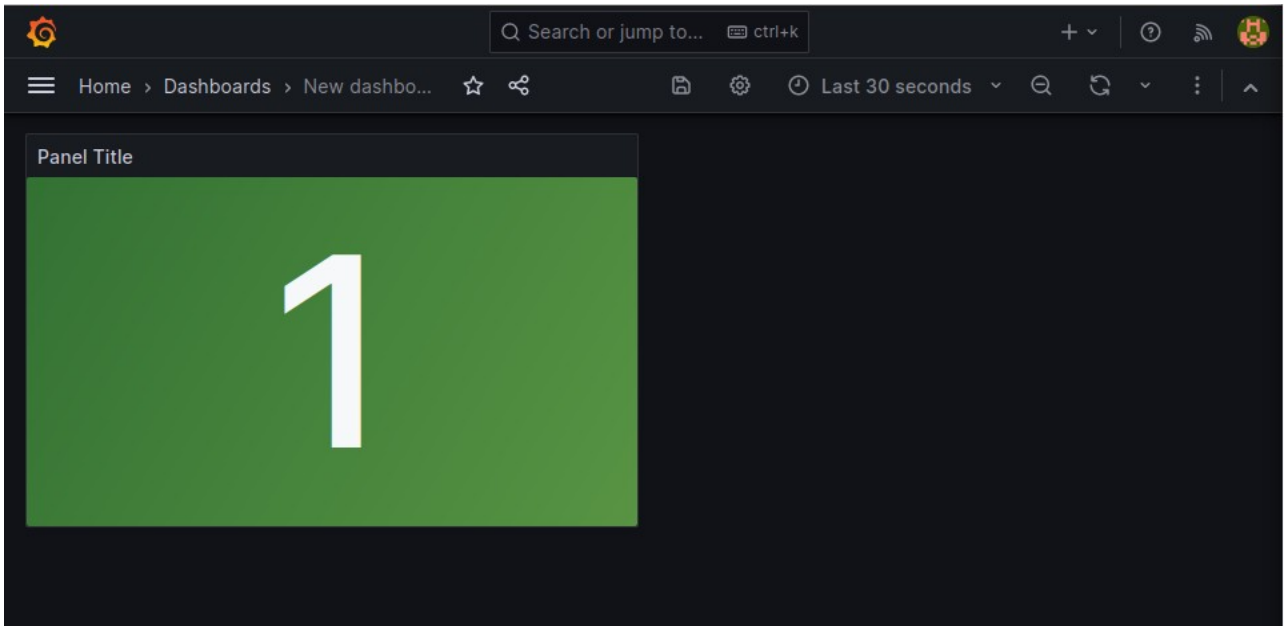
```
{{hostname}}-Node_exporter
```

« {{hostname}} » va chercher la valeur dans prometheus.yml qui est associé à chaque target, « -Node_exporter » peut être modifié selon vos besoins, vous pouvez tout à fait l'enlever si il vous déplaît.

Vous avez enfin terminé le paramètrage, vous pouvez donc sauvegarder et appliquer grâce aux deux boutons en haut à droite.

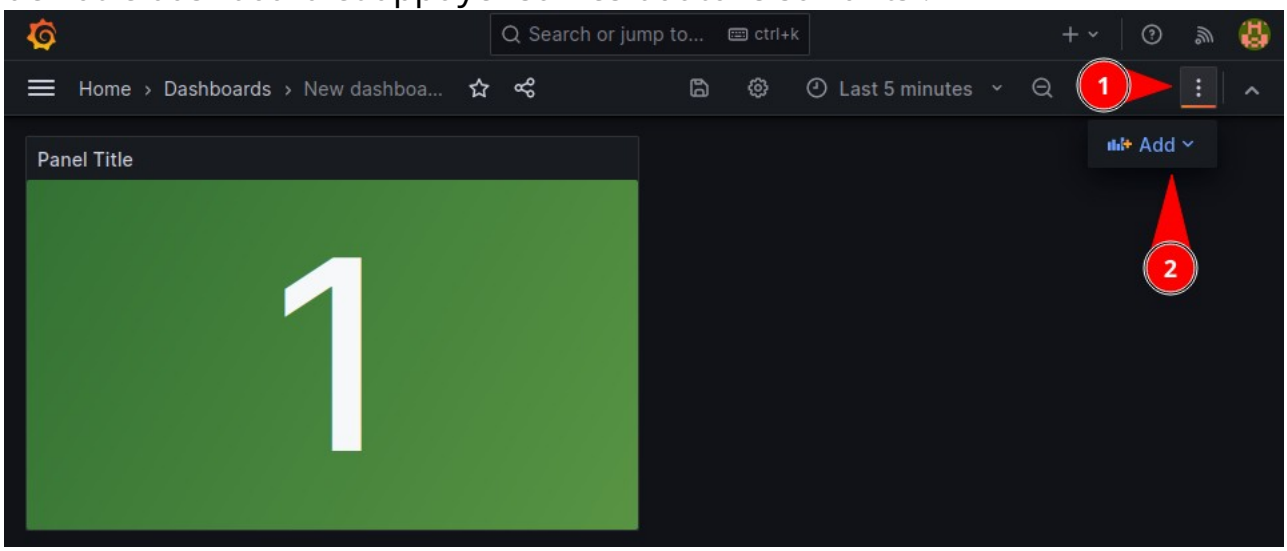


Bravo, vous avez enfin fini votre premier graphique, mais il faut avouer que savoir que votre Node exporter fonctionne ne va pas vous apportez grand-chose sans d'autres informations.



4.1.2 – Surveillance usage de la RAM / du CPU / du HDD

Pour créer un nouveau graphique (pas dashboard), mettez vous sur l'écran de votre dashboard et appuyer sur les boutons suivants :



Pour cette partie, je ne fournirais que les requêtes car nous avons déjà vu le menu de réglages des métriques avec la mise en place du graphique de **up {job= »node »}** et que je ne peux expliquer toutes les fonctionnalités de Grafana, je vous invite donc à fouiller par vous même ce qui est possible en terme de graphique.

Requête pour le taux d'utilisation de la RAM :

```
100 * (1 - ((avg_over_time(node_memory_MemFree_bytes[10m]) +
avg_over_time(node_memory_Cached_bytes[10m]) +
avg_over_time(node_memory_Buffers_bytes[10m])) /
avg_over_time(node_memory_MemTotal_bytes[10m])))
```

Requête pour le taux d'utilisation du processeur :

```
(1 - avg(irate(node_cpu_seconds_total{mode="idle"}[1m])) by (hostname)) * 100
```

Requête pour le taux de remplissage de la ROM (Disque dur) :

```
100 - ((node_filesystem_avail_bytes{mountpoint="/",fstype!="rootfs"} * 100) /
node_filesystem_size_bytes{mountpoint="/",fstype!="rootfs"})
```

Je vous recommande de créer un nouveau graphique pour chaque requête, si vous ne le faites pas, vous vous retrouverez avec l'ensemble de vos valeurs sur un seul graphique ; ce qui peut être très dur à lire.

4.1.3 – Surveillance des services

Surveiller le fonctionnement des machines c'est bien, surveiller en plus le fonctionnement de chaque service sur chaque machine c'est mieux.

Pour cela, vous devez rajouter au service de node exporter (le fichier qui permet de le lancer à chaque démarrage de votre machine) un paramètre en plus pour le démarrage :

```
nano /lib/systemd/system/<Nom du service>
```

Il n'est pas impossible que le fichier se trouve a un autre emplacement, si c'est le cas, vous pourrez le trouver en faisant :

```
sudo systemctl status <Nom du service>
```

```
root@debian:/etc/systemd/user# systemctl status prometheus-node-exporter
● prometheus-node-exporter.service - Prometheus exporter for machine metrics
     Loaded: loaded (/lib/systemd/system/prometheus-node-exporter.service; enabled; preset: enabled)
     Active: active (running) since Wed 2023-10-18 12:34:19 CEST; 4h 30min ago
       Docs: https://github.com/prometheus/node_exporter
   Main PID: 428 (prometheus-node)
      Tasks: 5 (limit: 495)
     Memory: 14.7M
        CPU: 26.070s
     CGroup: /system.slice/prometheus-node-exporter.service
             └─428 /usr/bin/prometheus-node-exporter
```

Vous devez donc rajouter le texte suivant au niveau de ExecStart :

```
[Unit]
Description=Prometheus exporter for machine metrics
Documentation=https://github.com/prometheus/node_exporter

[Service]
Restart=on-failure
User=prometheus
EnvironmentFile=/etc/default/prometheus-node-exporter
ExecStart=/usr/bin/prometheus-node-exporter --collector.systemd
ExecReload=/bin/kill -HUP $MAINPID
TimeoutStopSec=20s
SendSIGKILL=no

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

Puis rafraîchissez les daemons et redémarrer le service de Node_exporter :

```
sudo systemctl daemon-reload
sudo systemctl restart <Nom du service>
```

Il est nécessaire de faire cette modification pour chaque machine que l'on souhaite surveiller au niveau de ces services.

Vous pouvez enfin retourner sur le site de Grafana.

La requête pour surveiller le fonctionnement d'un service :

```
node_systemd_unit_state{hostname="<Nom de la machine>",name="<Nom du service>.service",state="active"}==1
```

Si vous n'avez pas configuré le hostname de vos targets dans 4.1.1, la requête risque de vérifier le fonctionnement du service sur toutes vos machines ; hors certaines d'entre elles n'ont pas forcément le service, ce qui peut poser problème car Grafana vous fera un retour comme quoi vos machines ont des problèmes de services alors qu'il n'en y a pas.

5 - Annexes

Fiche de recette

Vérification de l'opérationnalité de la solution mise en œuvre : *Grafana*

Description du test :

1. *Vérification du statut du service*
2. *Tentative de connexion à l'interface Web*
3. *Tentative de récupération de données à partir d'un dashboard*

Résultats Attendus :

1. *Service en active(running)*
2. *Accès à l'interface*
3. *Données récupérées*

Réception Globale : Grafana **Auteurs: Timothée LIGNIERE**	**Date: 12/10/23**
Reçu : ☐	
Reçu avec réserve : ☐ ……	
Refusé : ☐……	
Commentaire :	

Recette étape par étape * — ****** *(pour chaque étape, vous devez élaborer dans un fichier distinct un scénario détaillé à faire appliquer au « client » venant valider votre solution)*

Réception Etape 1: Faites « sudo systemtl status grafana-server » pour vérifier l'état du service, si il est bien fonctionnel, il devrait être en active(running)
Reçu : ☐
Reçu avec réserve : ☐……
Refusé : ☐……
Commentaire :

Réception Etape 2 : *Tenter de vous connectez à l'interface web en allant sur le port 3000 de votre machine à l'aide d'un navigateur*
Reçu : ☐
Reçu avec réserve : ☐……
Refusé : ☐……
Commentaire :

Réception Etape 3 : *Tenter de récupéré des données à l'aide d'un dashboard, il n'y a pas d'importance sur la donnée récupérée*
Reçu : ☐
Reçu avec réserve : ☐……

Refusé : ☐	……
Commentaire :	