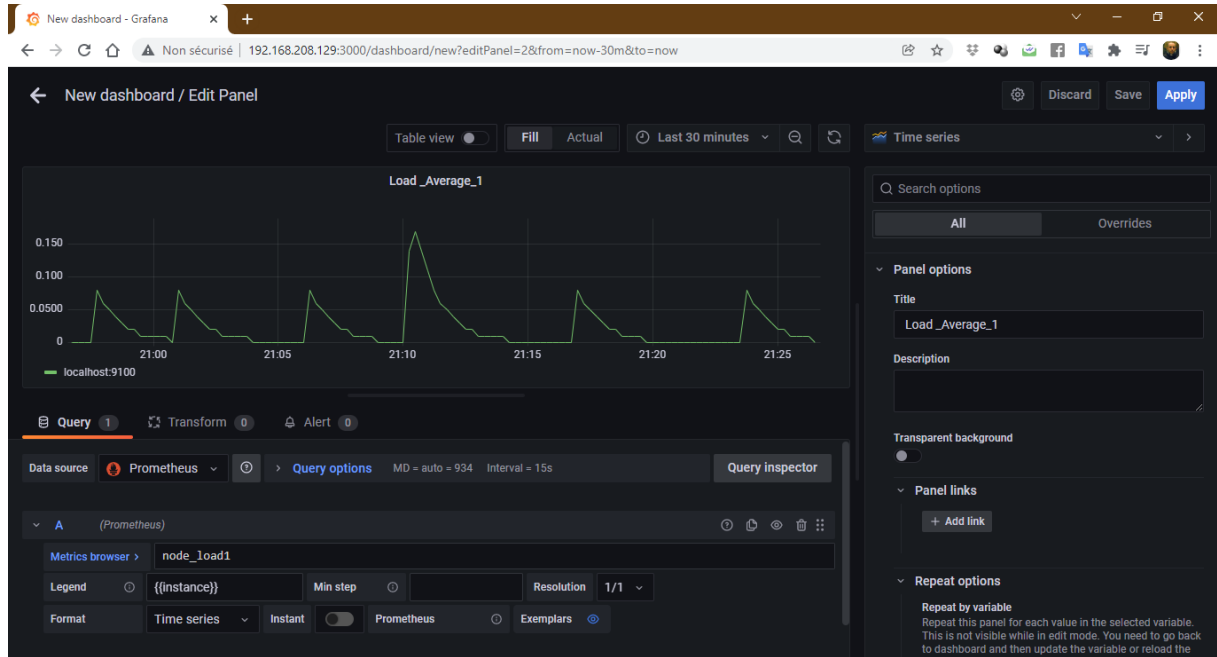


## Lab 5 –Grafana – Les graphiques

### Premier graphique de type « Graph »

Dans le menu principale de grafana, cliquez sur Create > Dashboard

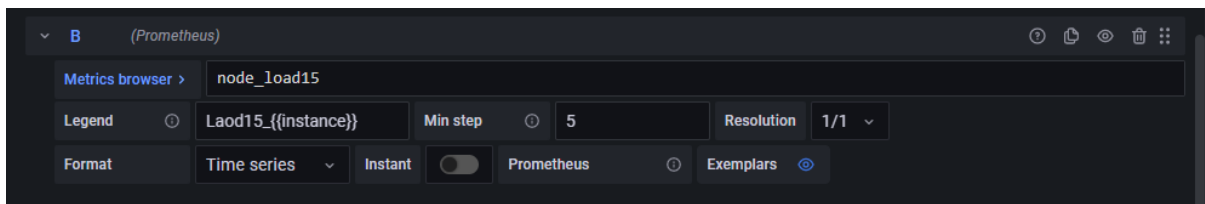
Puis ajoutez un nouveau Panel



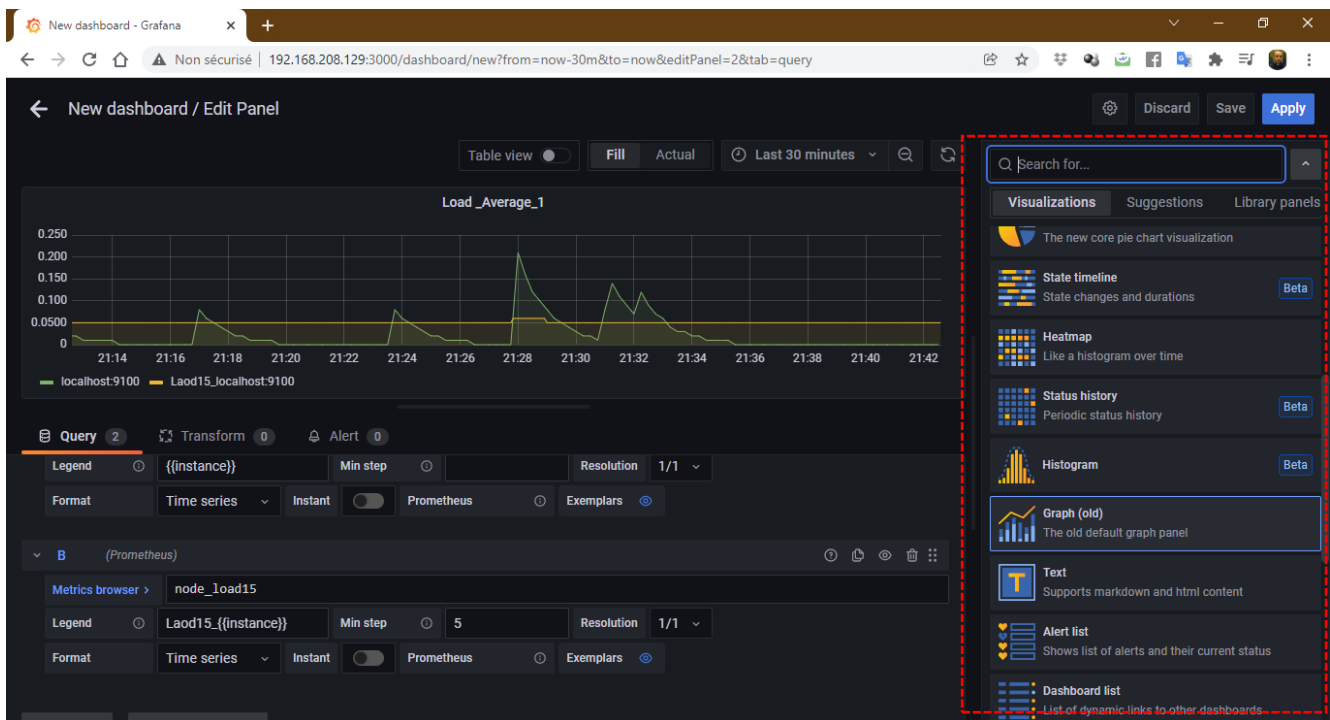
- **Datasource** : la source des données à utiliser
- **Query** : la requête PromQL à évaluer pour votre graphique
- **Legend** : personnaliser les légendes de graphique. Vous pouvez utiliser la syntaxe GO pour un affichage dynamique ; exemple {{instance}} pour afficher l'IP de l'instance
- **min step** : intervalle de temps
- **instant** : valeur instantanée
- **résolution** :
  - ✓ 1/1 : une valeur sur une affichée
  - ✓ 1/2 : une valeur sur deux...
- **prometheus** : lien vers la requête

### Modifier le graphique

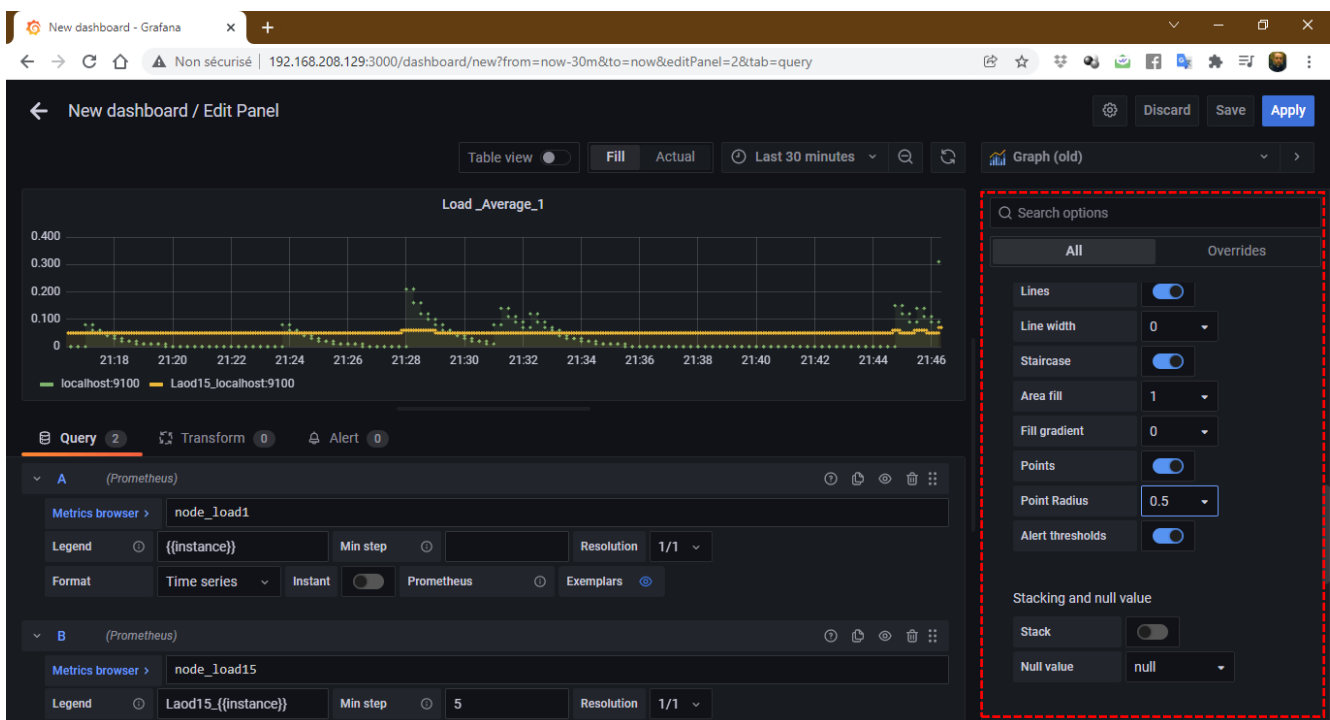
Ajoutez une nouvelle requête



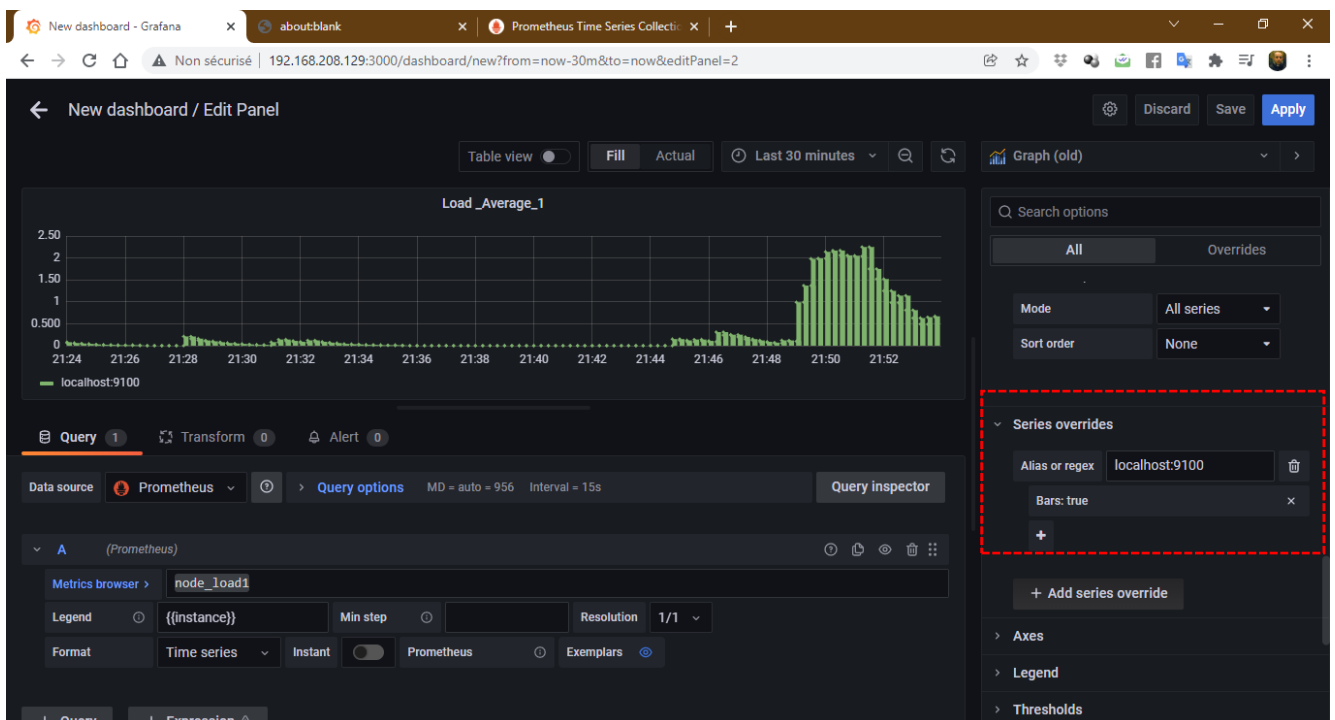
Modifiez le type de graphique



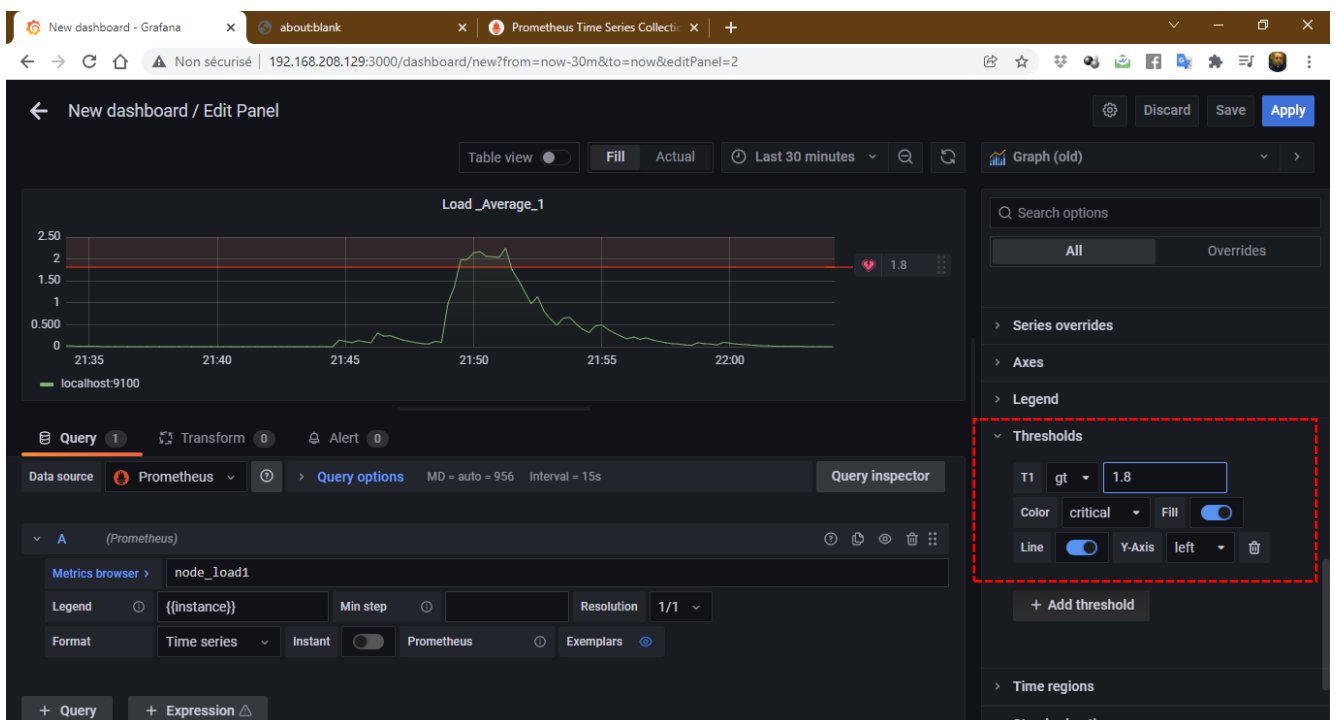
Modifiez les paramètres du Display



Ajoutez une « Series overrides » pour personnaliser l’affichage de certaines valeurs

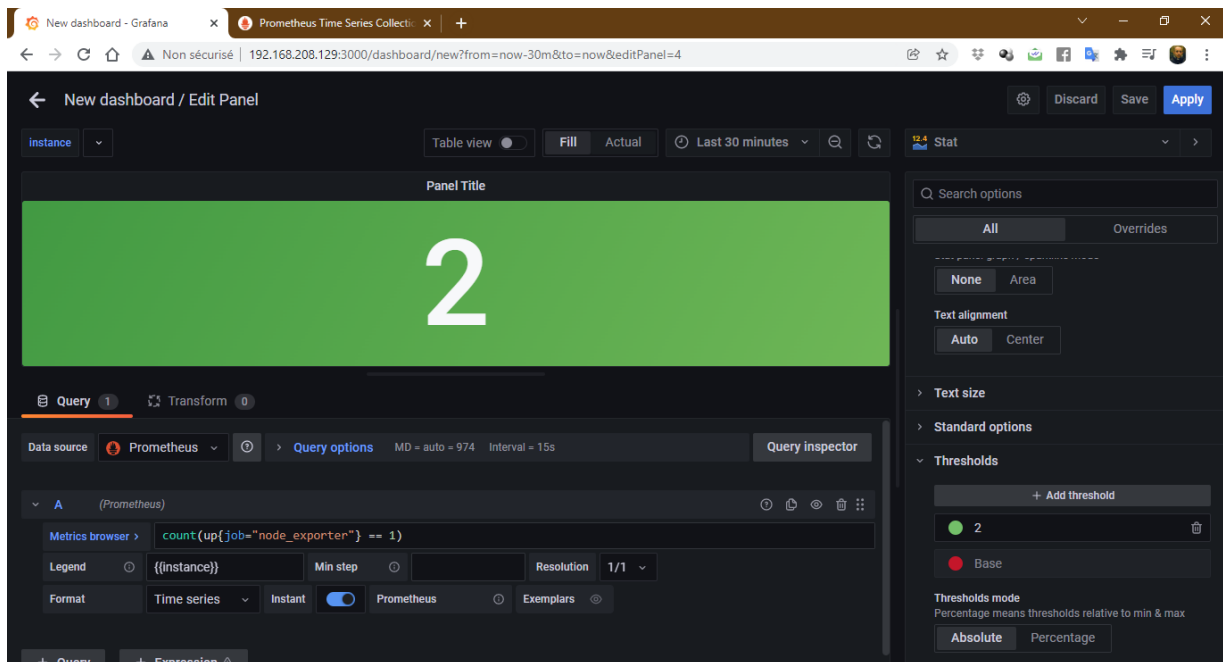


Ajoutez un seuil dans le graphique (exemple les valeurs qui dépasse le 1.8 du loadaverage sont dans un état critique)



## Graphique de type Stat

Dans cet exemple nous allons ajouter un graphique de Stat pour vérifier les instances qui sont actifs

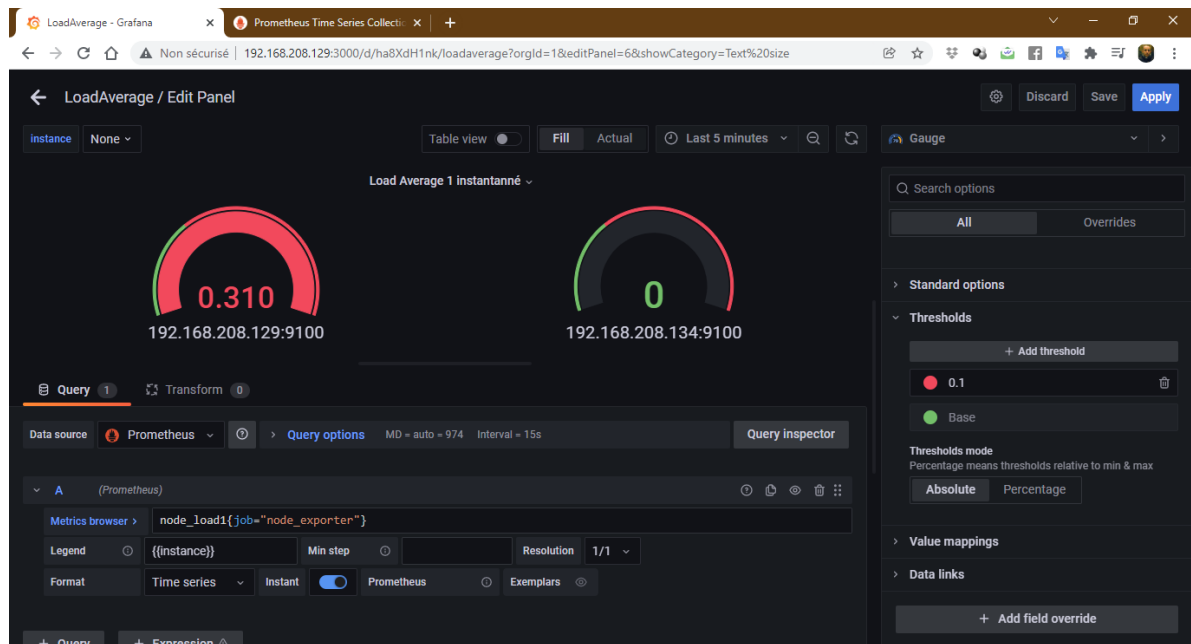


Ajustez les paramètres suivants :

- ✓ La couleur en arrière-plan
- ✓ Graph mode : none
- ✓ Ajoutez un seuil pour afficher l'arrière-plan en vert si le nombre d'instance up = 2, sinon rouge

## Graphique de type Gauge

Dans cet exemple nous allons afficher le load average de nos nodes instantané en graphique de type gauge.

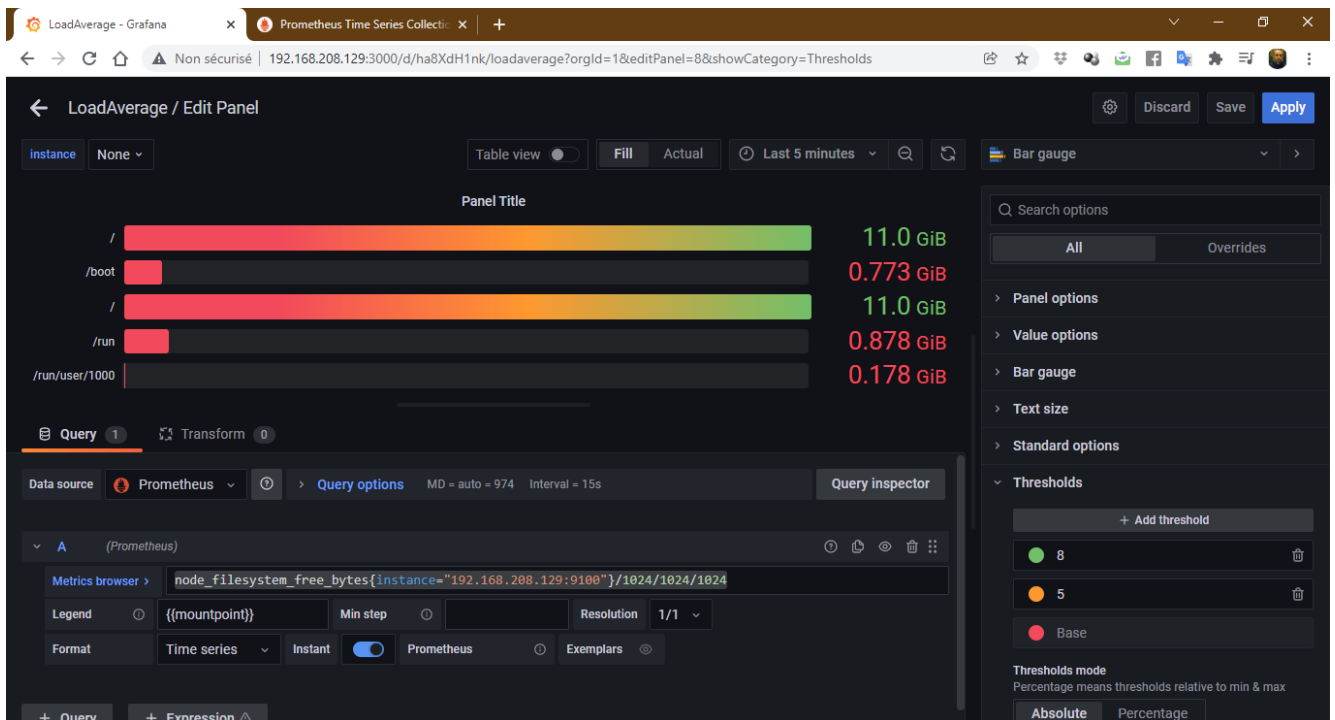


Ajustez les paramètres suivants :

- ✓ Insérez la requête suivante : `node_load1{job="node_exporter"}`
- ✓ Mode des valeurs : instantané
- ✓ Ajoutez un titre du Panel et une description
- ✓ Ajouter un seuil pour afficher en rouge si le load average dépasse le 0.18
- ✓ Mettez l'arrière-plan en transparent

## Graphique de type Bar Gauge

Dans cet exemple nous allons afficher l'occupation de notre système de fichier en utilisant le graphique de type Bar Gauge



Ajustez les paramètres suivants :

- ✓ Insérez la requête suivante :  
`node_filesystem_avail_bytes{instance="192.168.208.129:9100"}/1024/1024/1024`
- ✓ Mode des valeurs : instantané
- ✓ Ajoutez un titre du Panel et une description
- ✓ Mettez l'arrière-plan en transparent
- ✓ Ajouter un seuil :
  - En vert si l'espace disponible > 8GO
  - En Jaune si l'espace disponible > 5
  - Sinon en rouge
- ✓ Changez l'unité d'affichage en GiB

## Ajouter des variables

Les variables sont un moyen de créer des tableaux de bord Grafana dynamiques. Ils peuvent être utilisés pour sélectionner différentes instances de métriques. Par exemple, si vous recevez des métriques de plusieurs machines, des variables peuvent être utilisées pour créer des listes déroulantes pour sélectionner les métriques d'une machine. Ces variables sont utilisées dans la requête de source de données pour prendre en charge les modifications des métriques dans le tableau de bord. Ajoutons une variable pour les noms de VM afin que nous puissions sélectionner des métriques pour différentes VM.

Les variables peuvent être de différents types, tels que:

- Type de source de données: qui peut être utilisé pour changer dynamiquement la source de données dans les panneaux.

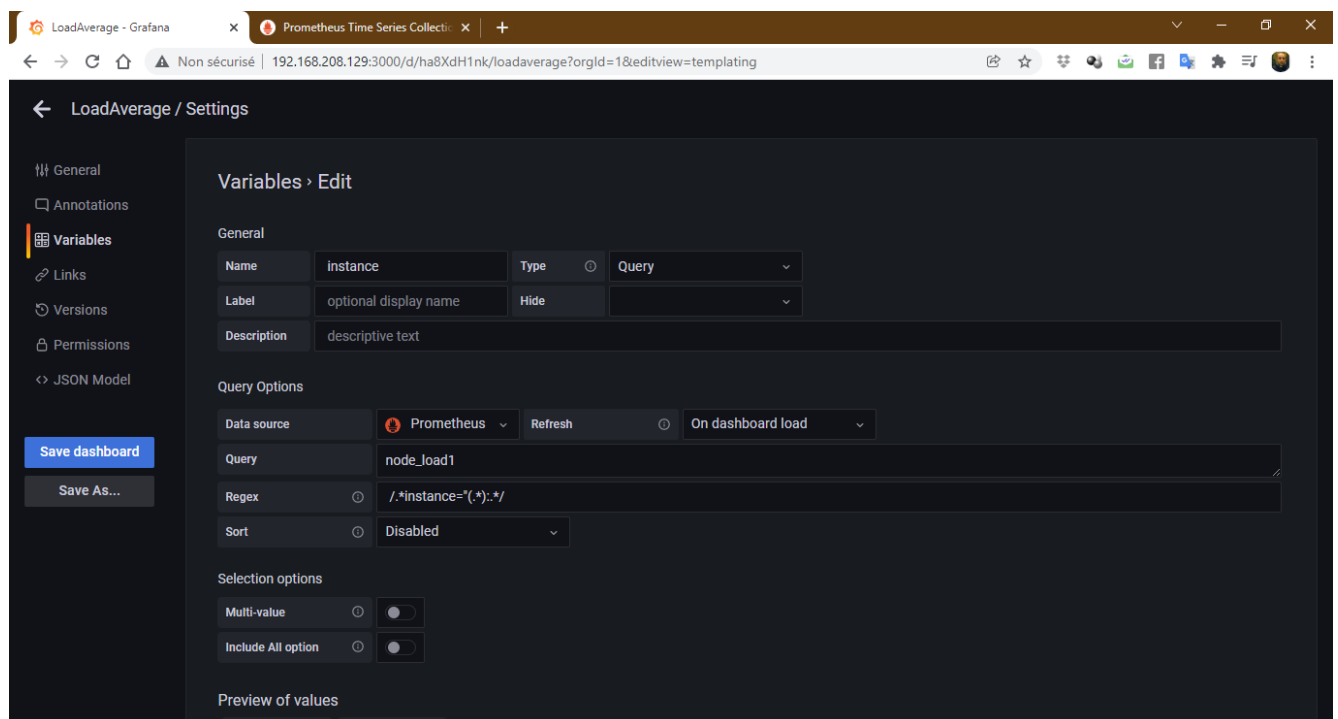
- Type de requête: où les valeurs telles que les noms d'instance et les hôtes du serveur peuvent être récupérées.
- Type d'intervalle: qui peut être utilisé pour effectuer une agrégation de manière dynamique sur les requêtes telles que l'utilisation du processeur et de la mémoire, de sorte que les derniers 1m, 5m, 10m peuvent être vus en utilisant des variables sans panneaux supplémentaires.

**Allez dans les paramètres du tableau de bord en haut à droite et cliquez sur variables → ajouter une nouvelle variable.**

Sélectionnez le type de requête et ajoutez la requête pour obtenir tous les noms d'hôte de l'exportateur de nœuds, que nous pouvons utiliser pour voir différentes statistiques de VM.

Pour extraire l'ip ou bien le nom de la machine, il faut préparer une expression régulière et capturer la séquence désirée, il faut mettre la section à capturer entre ().

- ✓ Name : instance
- ✓ Query : node\_load1
- ✓ Regex : `/.*/instance="(.*?)"/`



## Ajout de visualisation

Un panneau est le bloc de base de la visualisation dans Grafana. Il existe de nombreuses visualisations telles que Graph, Singlestat, Dashlist, Table, Text et plus encore si vous envisagez des plugins. Mais avant de créer ces visualisations, parlons de ce que nous voulons surveiller et des panneaux que nous utiliserons pour les visualisations.

Les mesures que vous choisissez de surveiller doivent répondre à deux questions: qu'est-ce qui est cassé et pourquoi?

Puisque nous voulons surveiller une machine virtuelle, nous devons surveiller les métriques de base telles que le processeur, la mémoire, le réseau, les E/S disque, l'espace disque, la disponibilité et le nombre de processus en cours d'exécution.

Si nous surveillions une application Web, nous voudrions surveiller le nombre de demandes entrantes, les temps de réponse, les codes de réponse, les ressources utilisées pour traiter une seule demande, les demandes en file d'attente et rejetées, etc.

Ajoutons maintenant quelques visualisations. Je vais expliquer quels panneaux utiliser pour surveiller les métriques: CPU, mémoire, système de fichiers, réseau.

1. CPU: Nous voulons voir l'utilisation actuelle du processeur et les heures supplémentaires d'utilisation du processeur. Pour voir l'utilisation actuelle du processeur, nous utiliserons la visualisation de type jauge. Une jauge est comme un compteur de vitesse, elle montera ou descendra dans une plage spécifique. Ajoutez un nouveau panneau et cliquez sur Ajouter une requête sur le panneau.
  - Le volet de droite affiche trois onglets. Dans l'onglet "Panel", Accédez à Visualisation et sélectionnez Gauge.
  - Sous Display, pour «Show», sélectionnez *Calculate*.
  - Pour *Calculate*, choisissez *Last* car nous voulons voir le dernier statut.
  - Dans l'onglet "Field" , sous "Standard Options", pour l'unité, sélectionnez le pourcentage (il existe de nombreux autres champs disponibles tels que KBps, miles, etc.)
  - Sous "Threshold", choisissez la valeur au-dessus de laquelle vous souhaitez afficher le rouge. Dans le cas ci-dessus, tout ce qui dépasse 80% d'utilisation du processeur sera affiché en rouge.

Pour visualiser les données, nous utiliserons `irate()` de `node_CPU` (utilisation moyenne du processeur), pour un intervalle de 5m, avec la variable d'instance comme `instance=~"$instance"`. Lorsque nous modifions l'instance à l'aide du menu déroulant, les mesures du panneau changent automatiquement.

```
100 - (avg(irate(node_cpu_seconds_total{mode="idle",
instance=~"$instance"}[5m]))*100)
```

- L'onglet Panel peut être utilisé pour modifier le titre du panneau, la description, les liens, etc. Ajoutons maintenant le titre en tant que CPU Usage [5m].
- Ajoutons le graphique chronologique de l'utilisation du processeur. Créez un nouveau panneau, ajoutez une requête et sélectionnez le type de visualisation sous forme de graphique



dans l'onglet de visualisation. Il existe de nombreuses options pour les types de graphiques tels que les graphiques à barres, les points, l'empilement, la largeur de ligne, les unités d'axe, la taille de l'échantillon, etc.

Grafana calcule automatiquement la durée des intervalles de temps dans chaque graphique. Il utilise les variables `$ _interval`, `$ _from` et `$ _to`. Par exemple, si vous visualisez les 6 derniers mois de données, il affichera des segments d'intervalle d'un jour dans le graphique, tandis que si vous visualisez les données de la dernière heure, il s'affichera dans des groupes d'intervalle de 1 m.

Maintenant que nous avons des visualisations de métriques CPU, ajoutons quelques métriques supplémentaires.

2. Mémoire ajoutons une visualisation de la mémoire totale et de l'utilisation actuelle de la mémoire. Nous ajouterons cette fois plusieurs requêtes pour visualiser la mémoire disponible et la mémoire totale.

```
node_memory_MemAvailable_bytes{instance="$instance:9100"}/1024/1024/1024
node_memory_MemTotal_bytes{instance="$instance:9100"}/1024/1024/1024
```

Ajouter un jauge pour la mémoire utilisé:

```
(node_memory_MemTotal_bytes{instance="$instance:9100"} -
node_memory_MemAvailable_bytes{instance="$instance:9100"} ) /
node_memory_MemTotal_bytes{instance="$instance:9100"} *100
```

3. Système de fichiers: Nous ajouterons le graphique de l'espace disque libre par rapport à l'espace disque total. Nous avons sélectionné les métriques `filesystem_free` et `filesystem_size` respectivement pour ce graphique. Ici, nous avons sélectionné le point de montage comme `/` pour visualiser les métriques pour le système de fichiers complet.

```
node_filesystem_free_bytes{instance="$instance:9100", mountpoint="/"}
node_filesystem_size_bytes{instance="$instance:9100", mountpoint="/"}
```

Précisez la légende comme: `free {{ mountpoint }}` et `total {{ mountpoint }}`

4. Réseau: dans cet exemple, nous visualisons la taille des données reçues et transmises sur différentes interfaces réseau. Ici, nous configurons l'unité de l'axe y en mégaoctets, mais elle peut être modifiée dans l'onglet visualisations en gigaoctets ou téraoctets (c'est beaucoup de trafic).

```
rate(node_network_receive_bytes_total{instance="$instance:9100"}[1m])
rate(node_network_transmit_bytes_total{instance="$instance:9100"}[1m])
```

Précisez la légende comme: **Received {{ device }}** et **Sent {{ device }}**

Faites maintenant glisser toutes les visualisations ajoutées vers le Row.

## Annotation

Les annotations permettent de marquer des points sur le graphique avec des événements riches. Lorsque vous survolez une annotation, vous pouvez obtenir une description de l'événement et des balises d'événement.

Le champ de texte peut inclure des liens vers d'autres systèmes avec plus de détails.

**Allez dans les paramètres du tableau de bord en haut à droite et cliquez sur annotations → ajouter une nouvelle annotation.**

Dans cet exemple nous allons ajouter une annotation sur la base de l'alerte Trop\_2\_load

```
- alert: Trop_2_load
  expr: node_load1 >= 0.6
  for: 20s
  labels:
    severity: critical
  annotations:
    summary: "{{ $labels.instance }}" - trop de load
    description: "Le server en prend plein la tronche "
```

Affecter cette configuration pour la configuration de l'annotation :

- **datasource** : prometheus
- **enabled**
- **Search expression** : ALERTS{alertstate="firing"}
- **Title** : {{alertname}} - {{instance}}
- **Tags** : incident

## Alerte

La visualisation des métriques est vraiment utile mais personne ne peut rester assis devant un ordinateur à regarder un tableau de bord toute la journée !!! Les alertes peuvent nous informer de mesures critiques telles qu'une utilisation élevée de la mémoire.

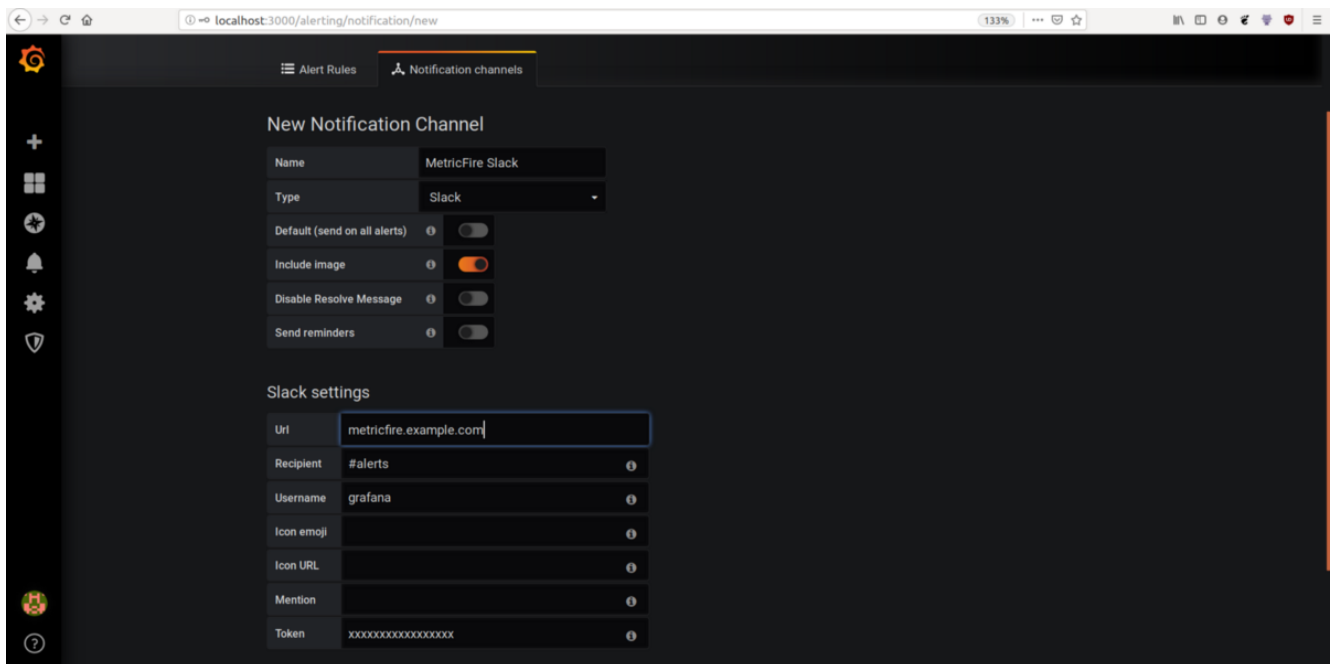
Configurons une alerte Slack sur le tableau de bord que nous venons de créer pour nous alerter si l'utilisation du processeur est élevée. Cliquez sur Alerte sur la page d'accueil et accédez aux canaux de notification et ajoutez un nouveau canal de notification pour Slack. Il nécessite des informations

d'identification, un nom de canal et un nom d'utilisateur slack.

[Visitez ce lien pour lier Slack à Grafana](#)

Après avoir configuré une destination d'alerte, nous allons accéder à l'icône d'alerte du panneau pour configurer les alertes. Nous allons configurer la condition suivante: si la valeur moyenne d'utilisation du processeur dans une plage de temps de 5m est supérieure à 90, l'alerte sera déclenchée. Nous

sélectionnons également où ces alertes doivent être envoyées, ici nous utilisons Slack que nous venons d'ajouter. Cette alerte est évaluée toutes les 1m. Si aucune donnée n'est disponible, nous configurons l'état sur alerte, ce qui signifie qu'il déclenchera l'alerte si aucune donnée n'est disponible.



The screenshot shows the 'New Notification Channel' configuration page in Grafana. The page is titled 'New Notification Channel' and has two tabs: 'Alert Rules' and 'Notification channels'. The 'Notification channels' tab is active. The form contains the following fields:

- Name:** MetricFire Slack
- Type:** Slack
- Default (send on all alerts):** ☐
- Include image:** ☒
- Disable Resolve Message:** ☐
- Send reminders:** ☐

Below these fields is the 'Slack settings' section with the following fields:

- Url:** metricfire.example.com
- Recipient:** #alerts
- Username:** grafana
- Icon emoji:**
- Icon URL:**
- Mention:**
- Token:** xxxxxxxxxxxxxxxxxxxx

Nous avons donc maintenant un tableau de bord fonctionnel et une configuration d'alerte.

## Importation et exportation de tableaux de bord

Chaque tableau de bord de Grafana est basé sur JSON. Ces tableaux de bord peuvent être exportés à partir d'un fichier JSON ou du référentiel de tableaux de bord Grafana .

Exportons le tableau de bord que nous venons de créer. Allez dans dashboard setting → JSON Model, il affiche un document JSON. Ce document JSON est une définition de tableau de bord complète qui peut être importée dans n'importe quelle autre instance Grafana. Sauvegardons-le sous node-exporter.json.

Voyons maintenant comment nous pouvons importer un tableau de bord. Accédez à l'icône + en haut à gauche et cliquez sur importer. Le JSON que nous avons créé ci-dessus peut être collé ci-dessous pour être importé, ou nous pouvons coller l'ID du tableau de bord à partir du référentiel de tableau de bord Grafana , et il sera importé.

Pour dupliquer un tableau de bord, accédez aux paramètres et enregistrez-le sous un nom différent et vous obtiendrez un tableau de bord en double.