



Monitoring Tools

Agenda

1. La Supervision
2. Prise de main avec Prometheus
3. PromQL
4. Alerting
5. Dashboarding avec Grafana

La Supervision

- Surveillance du bon fonctionnement d'un système ou d'une activité
- Surveiller, rapporter et aler les fonctionnements normaux et anormaux.
- Supervision : n.f, fonction qui consiste à indiquer et à commander l'état d'un appel, d'un système ou d'un réseau.



La Supervision

La surveillance informatique, la supervision, couvre :

- La disponibilité des serveurs, réseaux, logiciels, processus, switchs, routeurs, parefeux, onduleurs, connexions internet, bornes wifi...
- La disponibilité des ressources d'un système, telles que l'espace disque, la mémoire, le CPU,
- La santé des équipements (postes de travail, téléphonie, ...),
- La performance, par exemple les temps de réponse d'une application, le débit réseau, la température d'une salle,
- La fiabilité / qualité, par analyse de la disponibilité sur une période,
- La sécurité, pour prévenir des attaques, (vol de données confidentielles clients ou internes, ransomware, virus...)

La Supervision

- **La supervision technique** : consiste à surveiller le réseau, l'infrastructure et les machines du système d'information (Processeur, Mémoire, Stockage)
- **La supervision Applicative et Métier** : consiste à surveiller les applications et les processus métiers de l'entreprise.

La Supervision

- Les systèmes d'information sont tous différents de par leur **taille**, leur **nature**, leur **criticité**.
- Ils ont cependant pour point commun d'être le théâtre d'incidents, à un moment ou à un autre.
- La loi de Murphy est immuable. « Si quelque chose peut mal tourner, alors elle finira infailliblement par mal tourner ».
- Les administrateurs doivent concevoir l'architecture du système d'information de telle manière qu'une panne ait un **impact minimal** sur le reste du système.
- Ils doivent aussi gérer les éventuels problèmes.
- **80% de l'activité d'un administrateur consiste en la résolution de problèmes que rencontrent les utilisateurs.**

La Supervision

- La détection au plus tôt d'un dysfonctionnement, ou d'un risque, conduit à une résolution plus rapide, et limite les impacts pour l'entreprise et ses utilisateurs.
- Les mesures sont prises en amont, l'environnement est sécurisé et l'efficacité est améliorée.
- La supervision est le moyen de surveiller infrastructure et réseaux pour se prémunir efficacement des pannes.
- Ces données mettent en lumière les risques potentiels pour lesquels l'anticipation doit être l'assurance d'une exploitation optimale du SI.

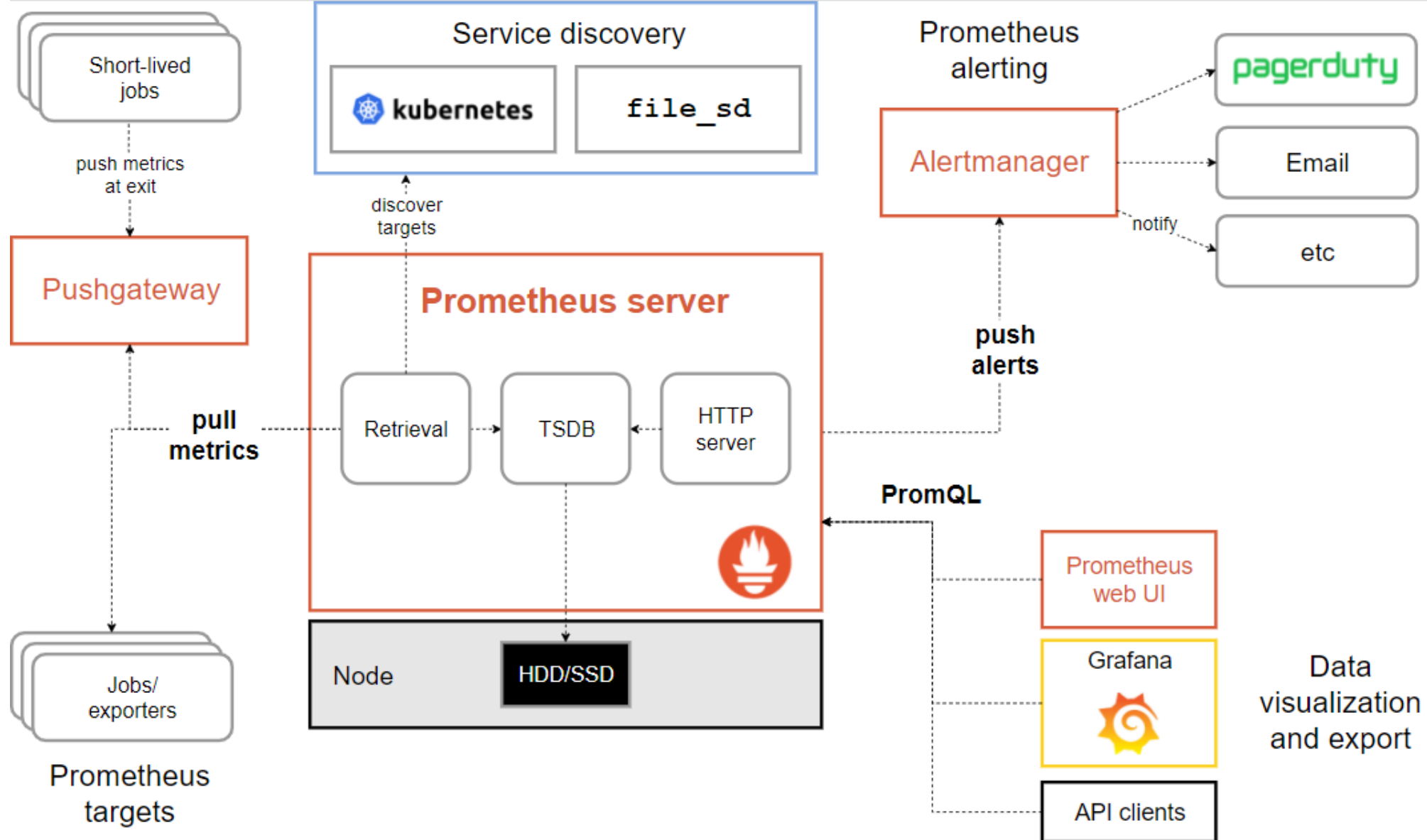
Qu'est ce que Prometheus ?

- Prometheus est une boîte à outils de surveillance et d'alerte de systèmes open source créée à l'origine chez **SoundCloud**.
- Prometheus doit son inspiration au Borgmon de Google.
- Il a été développé à l'origine par Matt T.Proud, un ancien SRE de Google
- Accepté en 2016 comme [deuxième projet de la CNCF \(Cloud Native Computing Foundation\)](#)

Qu'est ce que Prometheus ?

- Depuis sa création en 2012, de nombreuses entreprises et organisations ont adopté Prometheus, et le projet a une communauté de développeurs et d'utilisateurs très active.
- Il s'agit désormais d'un projet open source autonome et géré indépendamment de toute entreprise.

Architecture de Prometheus



Le Monitoring

- La majorité du Monitoring concerne la même chose: les **événements**.
- Les événements peuvent être presque n'importe quoi, y compris:
 - Envoi d'une réponse HTTP 400
 - Accès à une fonction
 - Atteindre le SINON d'une instruction SI
 - Quitter une fonction
 - Un utilisateur se connectant
 - Écriture de données sur disque
 - Lecture des données du réseau
 - Demander plus de mémoire au noyau
 - ...

Client Libraries

- Pour les applications propriétaires ou Open Source
- Quelqu'un doit introduire le code source qui produit leurs métriques
- Le projet **Prometheus** fournit des bibliothèques clientes officielles pour les langages **Go, Python, Java** et **Ruby**
- Il existe également une variété de bibliothèques clientes sous forme de tierce partie, pour d'autres langages tels que C#/.Net, Node.js, Haskell, Erlang et Rust.

Exporters

- Pour les applications fermées qu'on ne peut pas leur ajouter des instruments de mesure. (MySQL, Bind, Apache ..)
- Ces applications en général possèdent des interfaces à travers lesquelles on peut obtenir des mesures en formats spécifiques.
- Ces formats ne sont pas forcément compatibles avec Prometheus.
- Un **exportateur** est un logiciel que vous déployez juste à côté de l'application à partir de laquelle vous souhaitez obtenir des mesures.
- Il prend les requêtes de Prometheus, rassemble les données requises de l'application, les transforme dans le format correct, et les renvoie finalement en réponse à Prometheus.

Service Discovery

- Une liste statique de ressources fournie par l'utilisateur.
- Découverte basée sur des fichiers
 - par exemple, en utilisant un outil de gestion de configuration pour générer une liste de ressources qui sont automatiquement mises à jour dans Prometheus.
- Découverte automatisée
 - par exemple, interroger un repo de données comme Consul, exécuter instances dans Amazon ou Google, ou en utilisant des enregistrements DNS SRV pour générer une liste de ressources.

Scraping

- La **découverte** et le **réétiquetage** des services nous donnent une liste de cibles à surveiller.
- Maintenant, **Prometheus** doit récupérer les **métriques**.
- **Prometheus** le fait en envoyant une requête HTTP appelée **scrape**.
- La réponse au **scrape** est analysée et ingérée dans le stockage.
- Plusieurs mesures utiles sont également ajoutées, comme si le **scaping** a réussi et combien de temps cela a pris.
- Les **time-series** se produisent régulièrement; en général, vous le configurez pour qu'il se produise toutes les 10 à 60 secondes pour chaque cible.

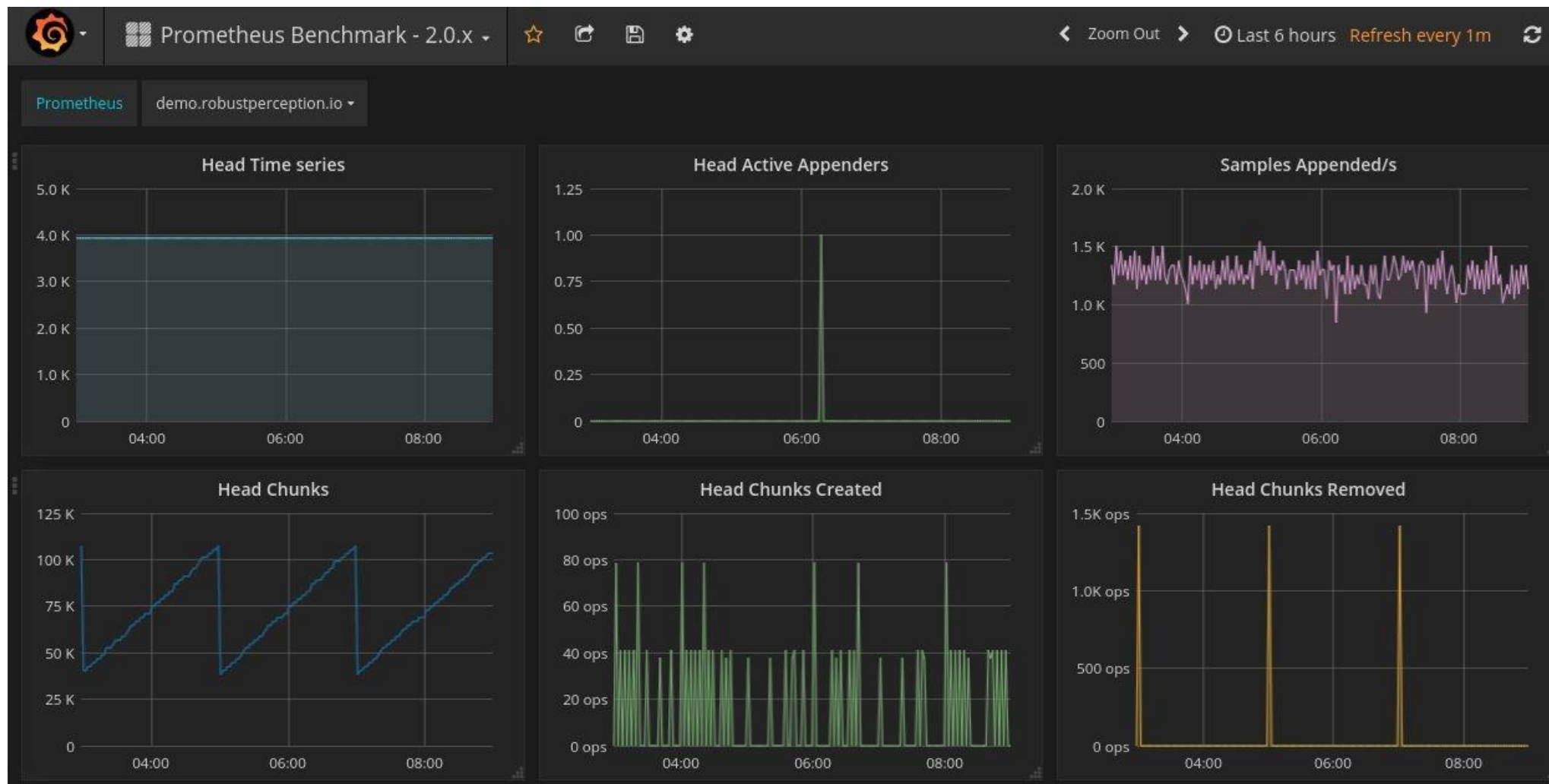
Storage

- **Prometheus** stocke les données localement dans une base de données personnalisée.
- Les systèmes distribués sont difficiles à rendre fiables, donc **Prometheus** n'essaye pas de faire une forme de **clustering**. En plus de la fiabilité, cela rend **Prometheus** plus facile à s'exécuter.
- Le système de stockage peut gérer l'ingestion de millions d'échantillons par seconde, ce qui permet de surveiller des milliers de machines avec un seul serveur **Prometheus**.
- L'algorithme de compression utilisé peut atteindre 1,3 octet par échantillon sur des données réelles.
- Un SSD est recommandé, mais pas strictement requis.

Dashboards

- API HTTP qui fournit des données brutes et évalue des requêtes **PromQL**.
- PromQL est utilisé pour produire des graphiques et des tableaux de bord.
- **Expression Browser** pour l'interrogation ad-hoc et à l'exploration de données.
- Il est recommandé d'utiliser **Grafana** pour les tableaux de bord.

Dashboards



Rules and Alerts

- Les **règles stockage** permettent d'évaluer régulièrement les expressions **PromQL** et d'intégrer leurs résultats dans le moteur de stockage.
- Les **règles d'alerte** sont une autre forme de **règles de stockage**.
- Ils évaluent également régulièrement les expressions **PromQL** et les résultats de ces expressions deviennent des **alertes**.
- Les alertes sont envoyées à **Alertmanager**.

Alert Management

- **Alertmanager** reçoit des alertes des serveurs **Prometheus** et les transforme en **notifications**.
- Les **notifications** peuvent inclure des e-mails, des applications de chat telles que **Slack** et des services tels que **PagerDuty**.
- **Alertmanager** fait plus que transformer aveuglément les alertes en notifications sur une base individuelle.
- Les alertes associées peuvent être regroupées en une seule notification.
- Les alertes peuvent également être mises en silence, peut-être pour un problème dont vous êtes déjà informé à l'avance lorsque vous savez que la maintenance est planifiée.
- Le rôle **d'Alertmanager** s'arrête à l'envoi de notifications;
- pour gérer les réponses humaines aux incidents, vous devez utiliser des services tels que **PagerDuty** et les systèmes de **Tickets**.

Prise de main avec Prometheus

set up and run Prometheus, the Node exporter, and the Alertmanager

Lancer Prometheus

1

Allez sur la page <https://prometheus.io/download/> et téléchargez la dernière version de Prometheus pour le système d'exploitation Linux avec Arch amd64

prometheus

The Prometheus monitoring system and time series database. [prometheus/prometheus](#)

2.2.1 / 2018-03-13 [Release notes](#)

File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.2.1.darwin-amd64.tar.gz	darwin	amd64	25.15 MiB	70166d0ca2f77d788e3a6a528765c17132f8f89ae681783fe5f76ff314f89993
prometheus-2.2.1.linux-amd64.tar.gz	linux	amd64	25.21 MiB	ec1798dbda1636f49d709c3931078dc17eafef76c480b67751aa09828396cf31
prometheus-2.2.1.windows-amd64.tar.gz	windows	amd64	25.07 MiB	03cf9f24a160944333e4db4358182b9e2d713872d27f126f7493e574493ae2c2

Lancer Prometheus

- ② Extrayez l'archive tar sur la ligne de commande et placez-la dans son répertoire

```
hostname $ tar -xzf prometheus-*.linux-amd64.tar.gz  
hostname $ cd prometheus-*.linux-amd64/
```

Modifiez maintenant le fichier appelé **prometheus.yml** pour qu'il contienne le texte suivant:

```
global:  
  scrape_interval: 15s  
scrape_configs:  
  - job_name: prometheus  
    static_configs:  
      - targets:  
        - localhost:9090
```

Lancer Prometheus

- ③ Vous pouvez maintenant exécuter le binaire Prometheus avec

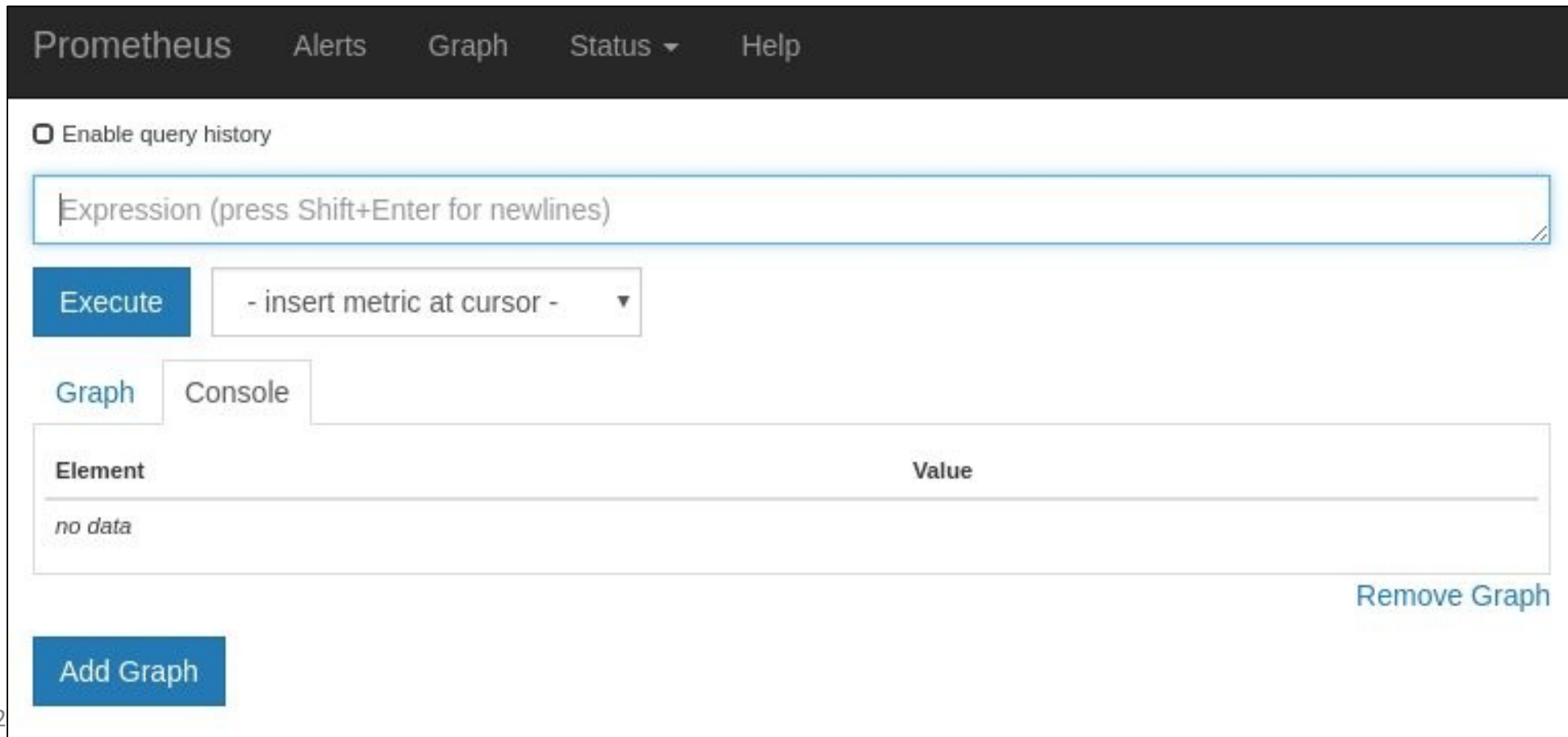
```
hostname $ ./prometheus
```



```
level=info ... msg="Starting Prometheus" version="(version=2.2.1, branch=HEAD,
revision=bc6058c81272a8d938c05e75607371284236aad)"
level=info ... build_context="(go=go1.10, user=root@149e5b3f0829, date=20180314-14:15:45)"
level=info ... host_details="(Linux 4.4.0-98-generic #121-Ubuntu..."
level=info ... fd_limits="(soft=1024, hard=1048576)"
level=info ... msg="Start listening for connections" address=0.0.0.0:9090
level=info ... msg="Starting TSDB ..."
level=info ... msg="TSDB started"
level=info ... msg="Loading configuration file" filename=prometheus.yml
level=info ... msg="Server is ready to receive web requests."
```


Lancer Prometheus

- 4 Vous pouvez maintenant accéder à l'interface utilisateur de Prometheus dans votre navigateur à **http: // localhost: 9090 /**, qui ressemblera à



The screenshot displays the Prometheus web interface. At the top is a dark navigation bar with links for 'Prometheus', 'Alerts', 'Graph', 'Status', and 'Help'. Below this, there is a checkbox labeled 'Enable query history'. A large text input field is labeled 'Expression (press Shift+Enter for newlines)'. Below the input field is a blue 'Execute' button and a dropdown menu currently showing '- insert metric at cursor -'. There are two tabs, 'Graph' and 'Console', with 'Graph' being the active tab. Below the tabs is a table with two columns: 'Element' and 'Value'. The table contains a single row with the text 'no data'. To the right of the table is a blue link labeled 'Remove Graph'. At the bottom left, there is a blue button labeled 'Add Graph'.

Element	Value
no data	

Utiliser l'Expression Browser

1

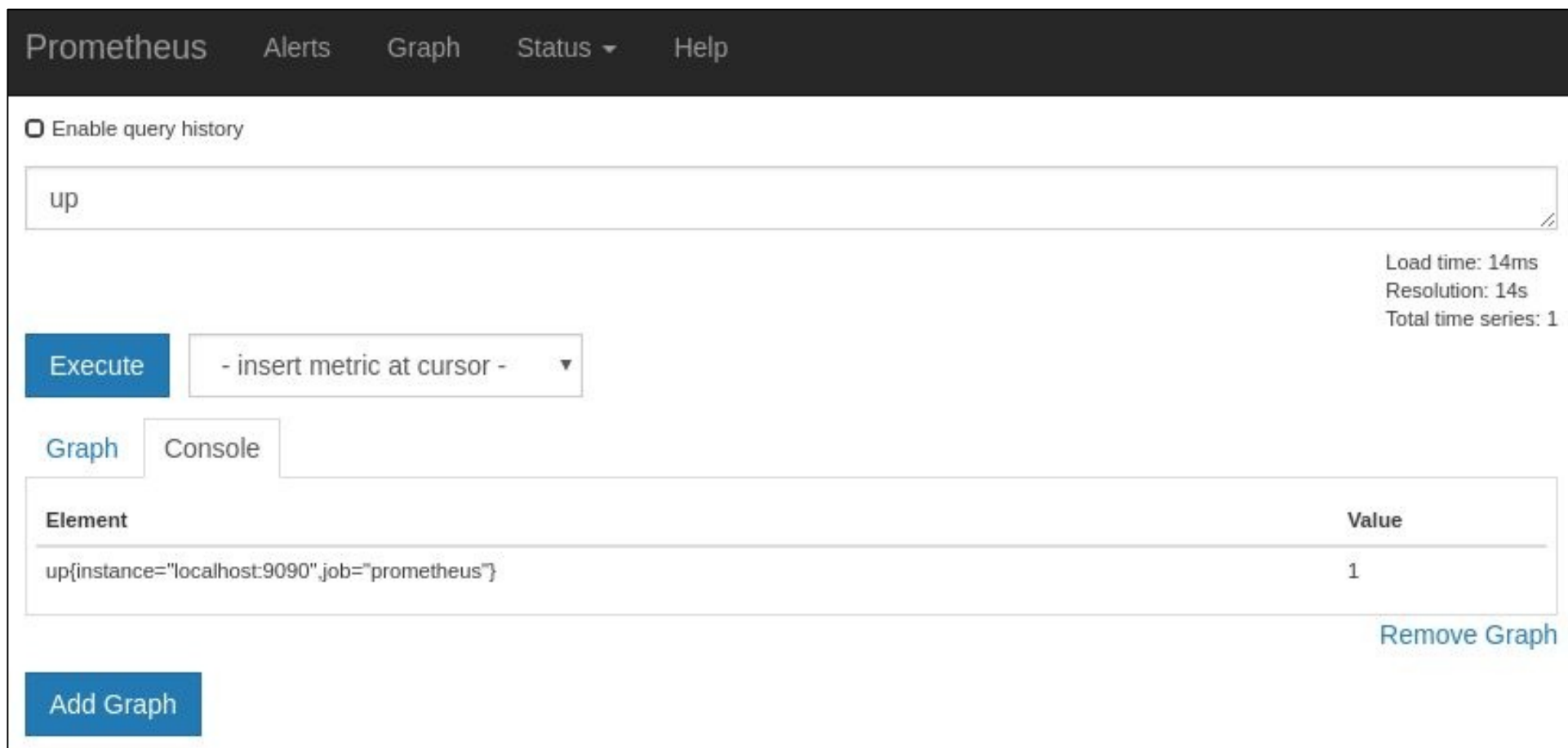
L'expression browser est utile pour exécuter des requêtes ad-hoc, développer des expressions PromQL et déboguer PromQL et les données à l'intérieur de Prometheus.

The screenshot shows the Prometheus Expression Browser interface. At the top is a dark navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this, there is a checkbox labeled 'Enable query history'. A large text input field is labeled 'Expression (press Shift+Enter for newlines)'. Below the input field are two buttons: 'Execute' and a dropdown menu currently showing '- insert metric at cursor -'. Below these are two tabs: 'Graph' and 'Console'. The 'Graph' tab is active, showing a table with two columns: 'Element' and 'Value'. The table contains a single row with the text 'no data'. To the right of the table is a link labeled 'Remove Graph'. At the bottom left is a blue button labeled 'Add Graph'.

Element	Value
no data	

Utiliser l'Expression Browser

- ② Pour commencer, assurez-vous que vous êtes dans la vue Console, entrez l'expression *up* et cliquez sur Exécuter.



The screenshot shows the Prometheus Expression Browser interface. At the top is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this is a section with a checkbox for 'Enable query history' and a text input field containing the query 'up'. To the right of the input field, performance metrics are displayed: 'Load time: 14ms', 'Resolution: 14s', and 'Total time series: 1'. Below the input field is a blue 'Execute' button and a dropdown menu currently showing '- insert metric at cursor -'. Underneath these are two tabs: 'Graph' and 'Console', with 'Console' being the active tab. The 'Console' view displays a table with two columns: 'Element' and 'Value'. The table contains one row with the element 'up{instance="localhost:9090",job="prometheus"}' and the value '1'. At the bottom right of the table is a 'Remove Graph' link. At the bottom left of the interface is a blue 'Add Graph' button.

Element	Value
up{instance="localhost:9090",job="prometheus"}	1

Utiliser l'Expression Browser

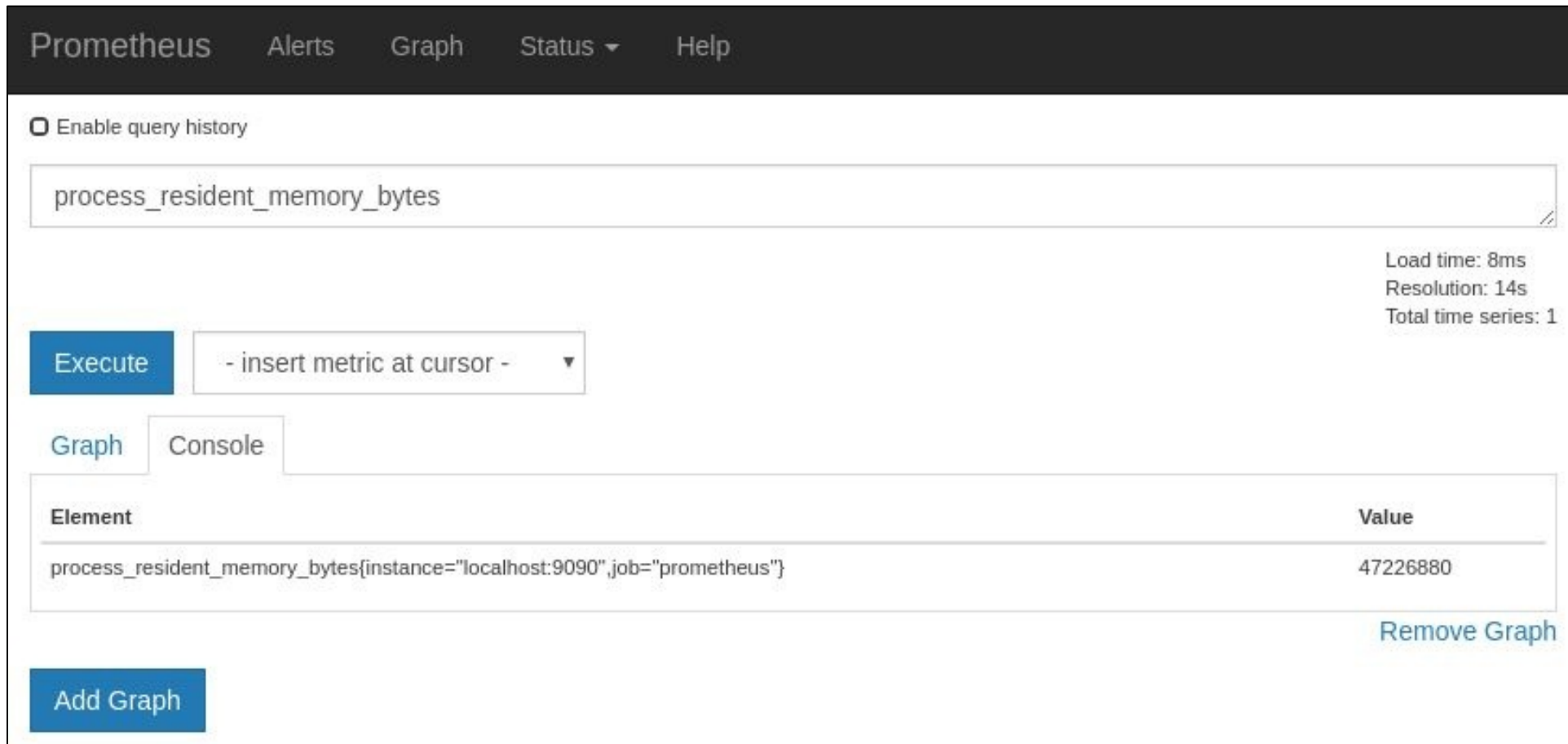
- ② Le résultat de la requête **up** est ici un seul job

```
up{instance="localhost:9090",job="prometheus"} 1
```

- **up** est une métrique spéciale ajoutée par **Prometheus** lors de l'exécution d'une **scrape**;
- **1** indique que la **scrape** a réussi;
- **instance** est une étiquette, indiquant **la cible** qui a été scrapé. Dans ce cas, cela indique que c'est Prometheus lui-même;
- L'étiquette **job** ici provient du **job_name** dans le fichier **prometheus.yml** pour indiquer le type du job.

Utiliser l'Expression Browser

3 Evaluer `process_resident_memory_bytes`



The screenshot shows the Prometheus Expression Browser interface. At the top, there is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this, there is a checkbox for "Enable query history". The main query input field contains the expression `process_resident_memory_bytes`. To the right of the input field, the following statistics are displayed: Load time: 8ms, Resolution: 14s, and Total time series: 1. Below the input field, there is a blue "Execute" button and a dropdown menu with the text "- insert metric at cursor -". Below the "Execute" button, there are two tabs: "Graph" and "Console". The "Console" tab is currently selected, showing a table with two columns: "Element" and "Value". The table contains one row with the element `process_resident_memory_bytes{instance="localhost:9090",job="prometheus"}` and the value `47226880`. Below the table, there is a blue "Add Graph" button and a link labeled "Remove Graph".

Prometheus Alerts Graph Status ▾ Help

☐ Enable query history

process_resident_memory_bytes

Load time: 8ms
Resolution: 14s
Total time series: 1

Execute - insert metric at cursor - ▾

Graph Console

Element	Value
process_resident_memory_bytes{instance="localhost:9090",job="prometheus"}	47226880

Remove Graph

Add Graph

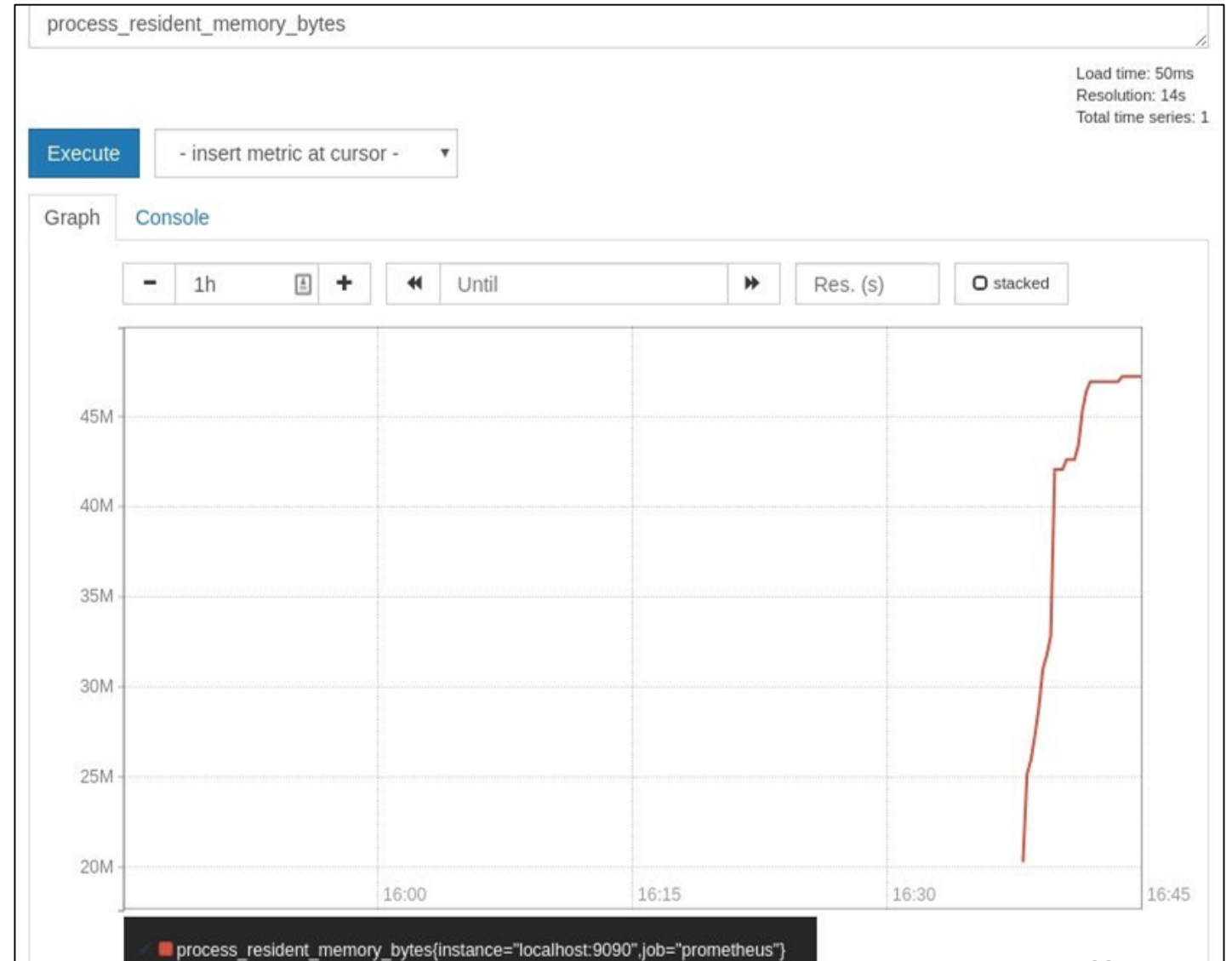
Utiliser l'Expression Browser

3

Cliquez sur **Graphique** pour passer à la vue graphique,

Pour voir comment elle a changé au fil du temps.

Ce type de métrique s'appelle **jauge**



Utiliser l'Expression Browser

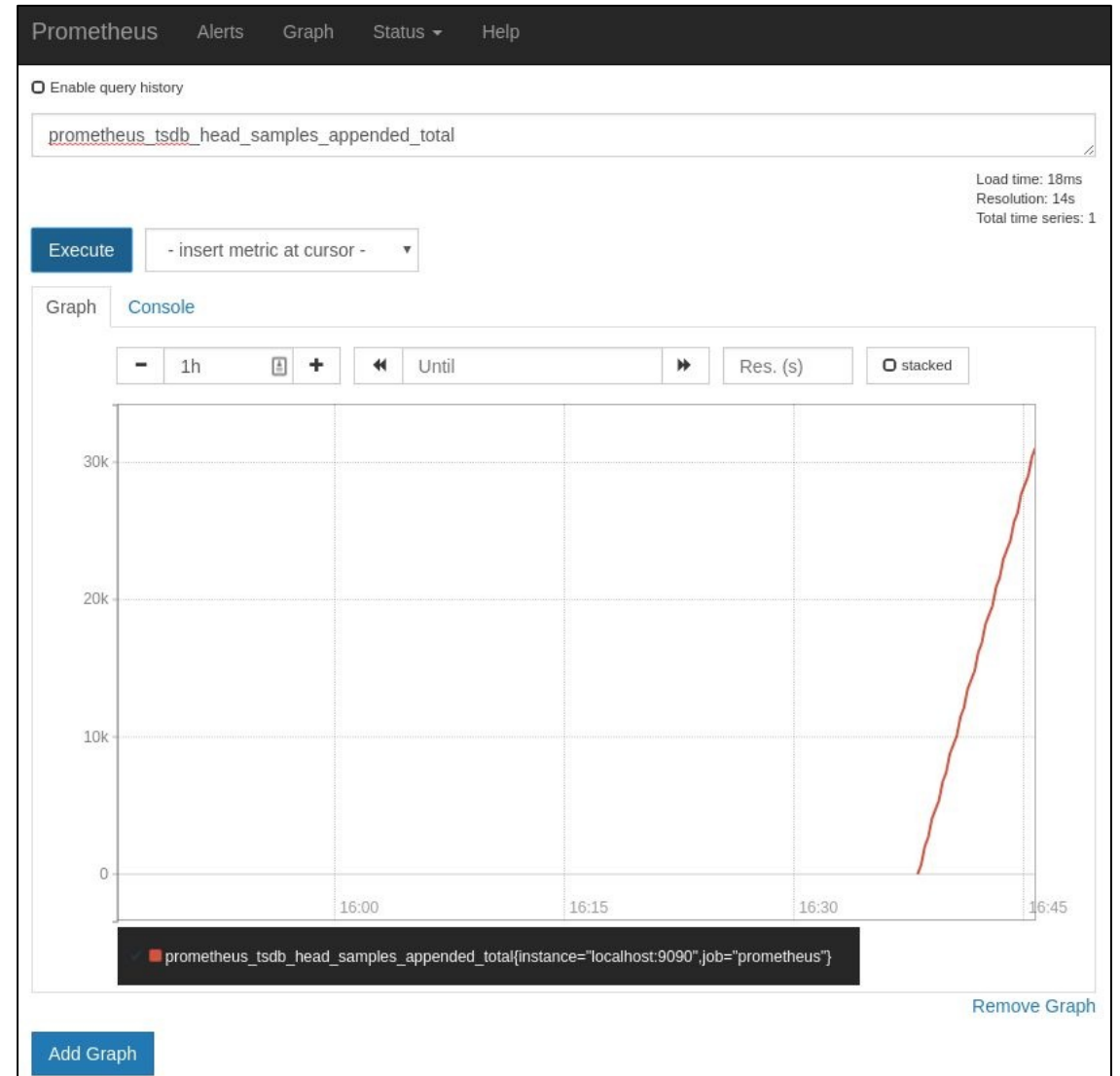
4

Il existe un deuxième type de métrique de base appelé **compteur**.

Les compteurs suivent le nombre d'événements qui se sont produits ou la taille totale de tous les événements.

Regardons un compteur représentant graphiquement le nombre d'échantillons que Prometheus a ingérés.

**`prometheus_tsdb_head_sample
s_appended_total`**



Utiliser l'Expression Browser

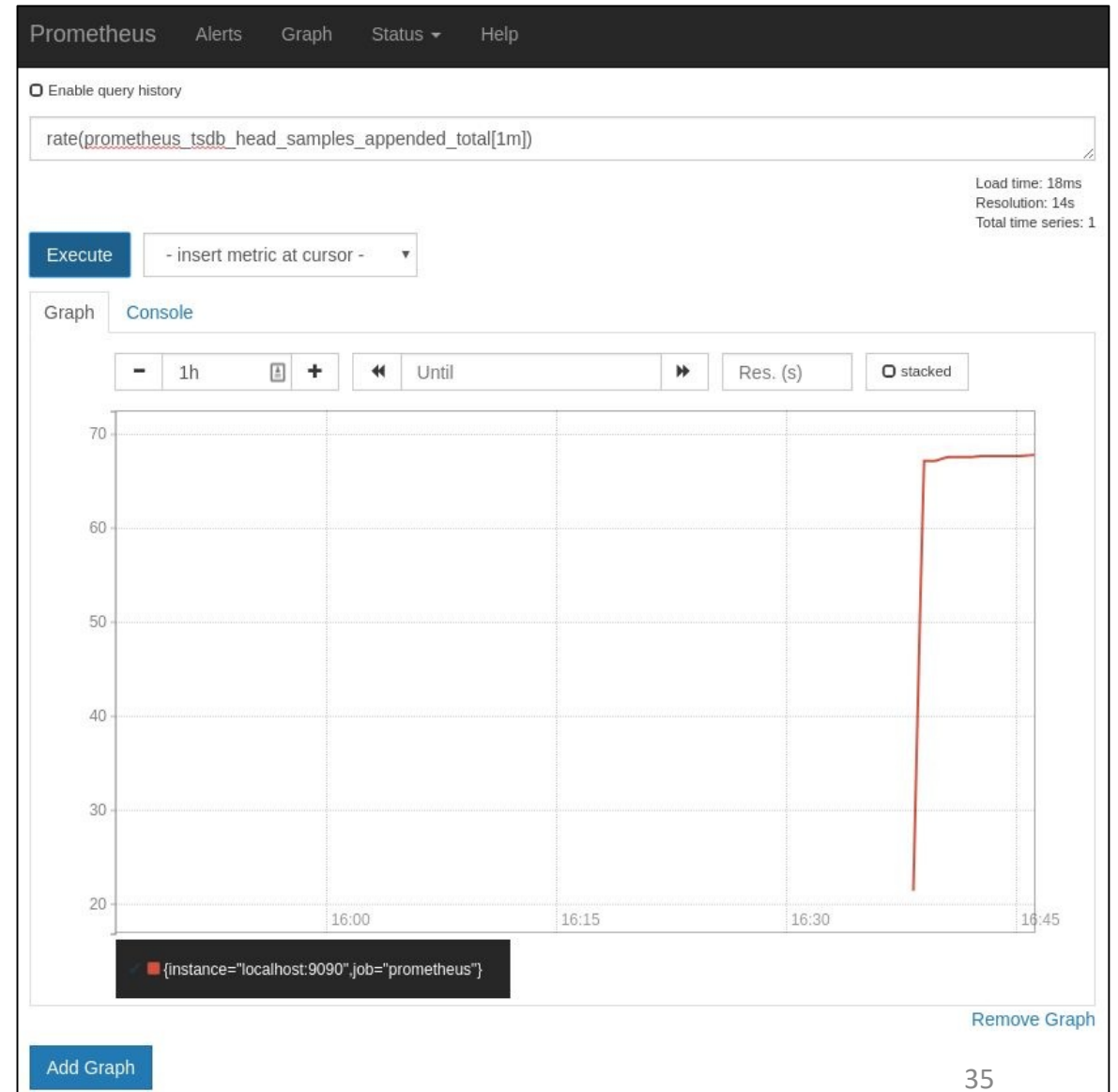
5

La fonction **rate** permet de savoir à quelle vitesse le compteur augmente.

La fonction **rate** calcule la vitesse à laquelle un compteur augmente par seconde.

Ajustez votre expression à
rate(prometheus_tsdb_head_samples_appended_total[1m])

qui calculera le nombre d'échantillons que Prometheus ingère par seconde en moyenne sur une minute



LAB 1 : Prometheus

Executer le Node Exporter

- ① Le node exporter expose les métriques au niveau du noyau et de la machine sur les systèmes Unix/Linux

Il fournit toutes les métriques standard telles que le processeur, la mémoire, l'espace disque, les E/S disque et la bande passante réseau

Ce que l'exportateur Node n'expose pas, ce sont

- les métriques sur les processus individuels,
- les métriques proxy d'autres exportateurs ou applications.

Executer le Node Exporter

- ① En raison de la grande variété de mesures disponibles à partir des systèmes d'exploitation, Node Exporter vous permet de configurer les catégories de mesures qu'il récupère.

Vous pouvez le faire avec des indicateurs de ligne de commande tels que :

- **--collector.wifi** activera le collecteur WiFi
- **--no-collector.wifi** le désactivera.

Executer le Node Exporter

① CPU Collector

La métrique principale du collecteur de CPU est *node_cpu_seconds_total*, qui est un compteur indiquant combien de temps chaque CPU a passé dans chaque mode. Les libellés sont *cpu* et *mode*.

```
# HELP node_cpu_seconds_total Seconds the cpus spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 48649.88
node_cpu_seconds_total{cpu="0",mode="iowait"} 169.99
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 57.5
node_cpu_seconds_total{cpu="0",mode="softirq"} 8.05
node_cpu_seconds_total{cpu="0",mode="steal"} 0
node_cpu_seconds_total{cpu="0",mode="system"} 1058.32
node_cpu_seconds_total{cpu="0",mode="user"} 4234.94
node_cpu_seconds_total{cpu="1",mode="idle"} 9413.55
node_cpu_seconds_total{cpu="1",mode="iowait"} 57.41
node_cpu_seconds_total{cpu="1",mode="irq"} 0
node_cpu_seconds_total{cpu="1",mode="nice"} 46.55
node_cpu_seconds_total{cpu="1",mode="softirq"} 7.58
node_cpu_seconds_total{cpu="1",mode="steal"} 0
node_cpu_seconds_total{cpu="1",mode="system"} 1034.82
node_cpu_seconds_total{cpu="1",mode="user"} 4285.06
```

Executer le Node Exporter

① CPU Collector

Pour chaque CPU, les modes augmenteront globalement d'une seconde par seconde.

Cela vous permet de calculer la proportion de temps d'inactivité sur tous les processeurs à l'aide de l'expression PromQL:

```
avg without(cpu, mode)(rate(node_cpu_seconds_total{mode="idle"}[1m]))
```

Cela fonctionne car il calcule le temps d'inactivité par seconde par CPU, puis fait la moyenne de tous les CPU de la machine.

Executer le Node Exporter

① FileSystem Collector

collecte des métriques sur vos systèmes de fichiers montés, comme la commande **df**.

Les indicateurs permettent de restreindre les systèmes de fichiers inclus:

- `--collector.filesystem.ignored-mount-points`
- et `--collector.filesystem.ignored-fs-types`

Toutes les métriques de ce collecteur sont préfixées par *node_filesystem_* et ont des étiquettes de *device*, *fstype* et de *mountpoint*.

```
# HELP node_filesystem_size_bytes Filesystem size in bytes.  
# TYPE node_filesystem_size_bytes gauge  
node_filesystem_size_bytes{device="/dev/sda5",fstype="ext4",mountpoint="/"} 9e+10
```

Executer le Node Exporter

① Diskstats Collector

Le collecteur **diskstats** expose les métriques d'E/S disque à partir de `/proc/diskstats`.

Par défaut, l'indicateur **--collector.diskstats.ignored-devices** tente d'exclure les éléments qui ne sont pas de vrais disques, tels que les partitions.

node_disk_io_now

The number of I/Os in progress.

node_disk_io_time_seconds_total

Incremented when I/O is in progress.

node_disk_read_bytes_total

Bytes read by I/Os.

node_disk_read_time_seconds_total

The time taken by read I/Os.

node_disk_reads_completed_total

The number of complete I/Os.

node_disk_written_bytes_total

Bytes written by I/Os.

node_disk_write_time_seconds_total

The time taken by write I/Os.

node_disk_writes_completed_total

The number of complete write I/Os.

Executer le Node Exporter

① NetDev Collector

Le collecteur **netdev** expose des métriques sur vos périphériques réseau avec le préfixe *node_network_* et une étiquette *device*.

```
# HELP node_network_receive_bytes_total Network device statistic receive_bytes.  
# TYPE node_network_receive_bytes_total counter  
node_network_receive_bytes_total{device="lo"} 8.3213967e+07  
node_network_receive_bytes_total{device="wlan0"} 7.0854462e+07
```

node_network_receive_bytes_total et *node_network_transmit_bytes_total* sont les principales métriques qui vous intéressent car vous pouvez calculer la bande passante du réseau avec elles:

rate(node_network_receive_bytes_total[1m])

Executer le Node Exporter

① Meminfo Collector

Le collecteur **meminfo** a toutes vos métriques standard liées à la mémoire avec un préfixe *node_memory_*.

Tout cela vient de votre /proc/meminfo, et c'est le premier collecteur où la sémantique devient un peu confuse.

Le collecteur convertit les kilo-octets en octets préférés:

```
# HELP node_memory_MemTotal_bytes Memory information field MemTotal.  
# TYPE node_memory_MemTotal_bytes gauge  
node_memory_MemTotal_bytes 8.269582336e+09
```

Executer le Node Exporter

① Meminfo Collector

- ***node_memory_MemTotal_bytes*** est la quantité totale de mémoire physique dans la machine.

Notez qu'il n'y a pas de métrique de mémoire utilisée, donc vous devez en quelque sorte la calculer.

- ***node_memory_MemFree_bytes*** est la quantité de mémoire qui n'est utilisée par aucun, mais cela ne veut pas dire que c'est toute la mémoire dont vous disposez.
- ***node_memory_MemAvailable*** est une heuristique du noyau pour la quantité de mémoire réellement disponible.

Executer le Node Exporter

2

Une version préconstruite de l'exportateur peut être téléchargée à partir de <https://prometheus.io/download/>.

Encore une fois, l'archive tar devra être extraite, mais aucun fichier de configuration n'est requis pour pouvoir être exécuté directement.

```
hostname $ tar -xzf node_exporter-*.linux-amd64.tar.gz
hostname $ cd node_exporter-*.linux-amd64/
hostname $ ./node_exporter
INFO[0000] Starting node_exporter (version=0.16.0, branch=HEAD,
revision=d42bd70f4363dced6b77d8fc311ea57b63387e4f) source="node_exporter.go:82"
INFO[0000] Build context (go=go1.9.6, user=root@a67a9bc13a69,date=20180515-15:52:42)
source="node_exporter.go:83"
INFO[0000] Enabled collectors: source="node_exporter.go:90"
INFO[0000] - arp source="node_exporter.go:97"
INFO[0000] - bacache source="node_exporter.go:97"
...
various other collectors
...
INFO[0000] Listening on :9100 source="node_exporter.go:111"
```

Executer le Node Exporter

② Lancer le Node Exporter avec `./node_exporter`

Vous pouvez accéder à l'exportateur dans votre navigateur à l'adresse `http://localhost:9100/` et visiter son endpoint `/metrics`.

Pour que Prometheus surveille l'exportateur, nous devons mettre à jour le fichier `prometheus.yml` en ajoutant une configuration de scrape supplémentaire:

```
global:
  scrape_interval: 10s
  scrape_configs:
    - job_name: prometheus
      static_configs:
        - targets:
            - localhost:9090
    - job_name: node
      static_configs:
        - targets:
            - localhost:9100
```

Executer le Node Exporter

- ③ Redémarrez Prometheus pour récupérer la nouvelle configuration en utilisant Ctrl-C pour le fermer puis redémarrez-le.
Si vous regardez la page Targets, vous devriez maintenant voir deux cibles, toutes deux à l'état **UP**

Prometheus Alerts Graph Status ▾ Help

Targets

☐ Only unhealthy jobs

node (1/1 up) [show less](#)

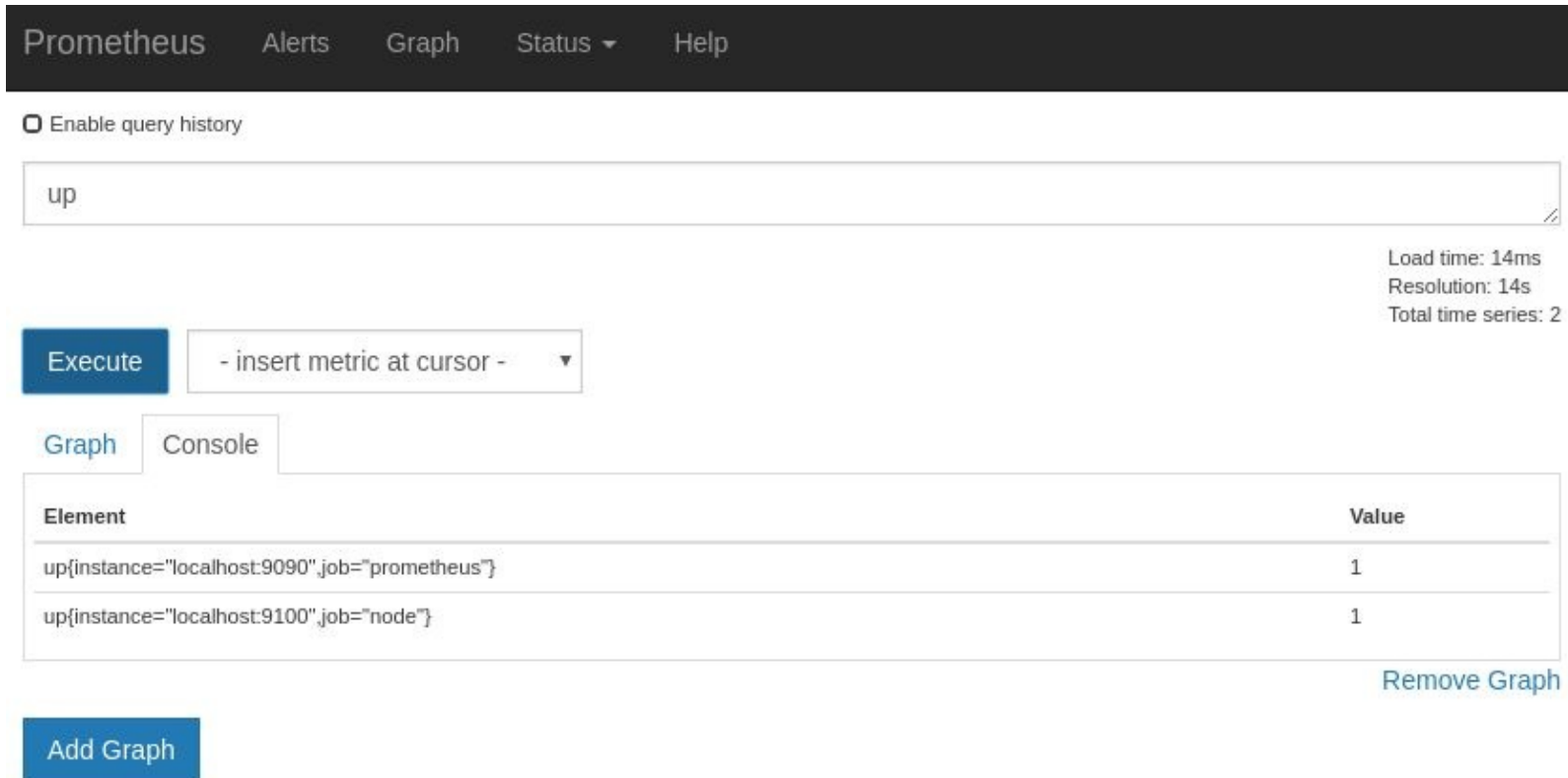
Endpoint	State	Labels	Last Scrape	Error
http://localhost:9100/metrics	UP	instance="localhost:9100"	4.936s ago	

prometheus (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"	6.094s ago	

Executer le Node Exporter

- 3 Vérifiez dans le Expression Browser que les deux entrées existent



The screenshot shows the Prometheus Expression Browser interface. At the top, there is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this, there is a checkbox for 'Enable query history'. The main query input field contains the text 'up'. To the right of the input field, the following statistics are displayed: Load time: 14ms, Resolution: 14s, and Total time series: 2. Below the input field, there is a blue 'Execute' button and a dropdown menu with the text '- insert metric at cursor -'. Below the 'Execute' button, there are two tabs: 'Graph' and 'Console'. The 'Graph' tab is selected, and it displays a table with two columns: 'Element' and 'Value'. The table contains two rows of data: 'up{instance="localhost:9090",job="prometheus"}' with a value of '1', and 'up{instance="localhost:9100",job="node"}' with a value of '1'. Below the table, there is a blue 'Add Graph' button and a link labeled 'Remove Graph'.

Prometheus Alerts Graph Status ▾ Help

☐ Enable query history

up

Load time: 14ms
Resolution: 14s
Total time series: 2

Execute - insert metric at cursor - ▾

Graph Console

Element	Value
up{instance="localhost:9090",job="prometheus"}	1
up{instance="localhost:9100",job="node"}	1

Remove Graph

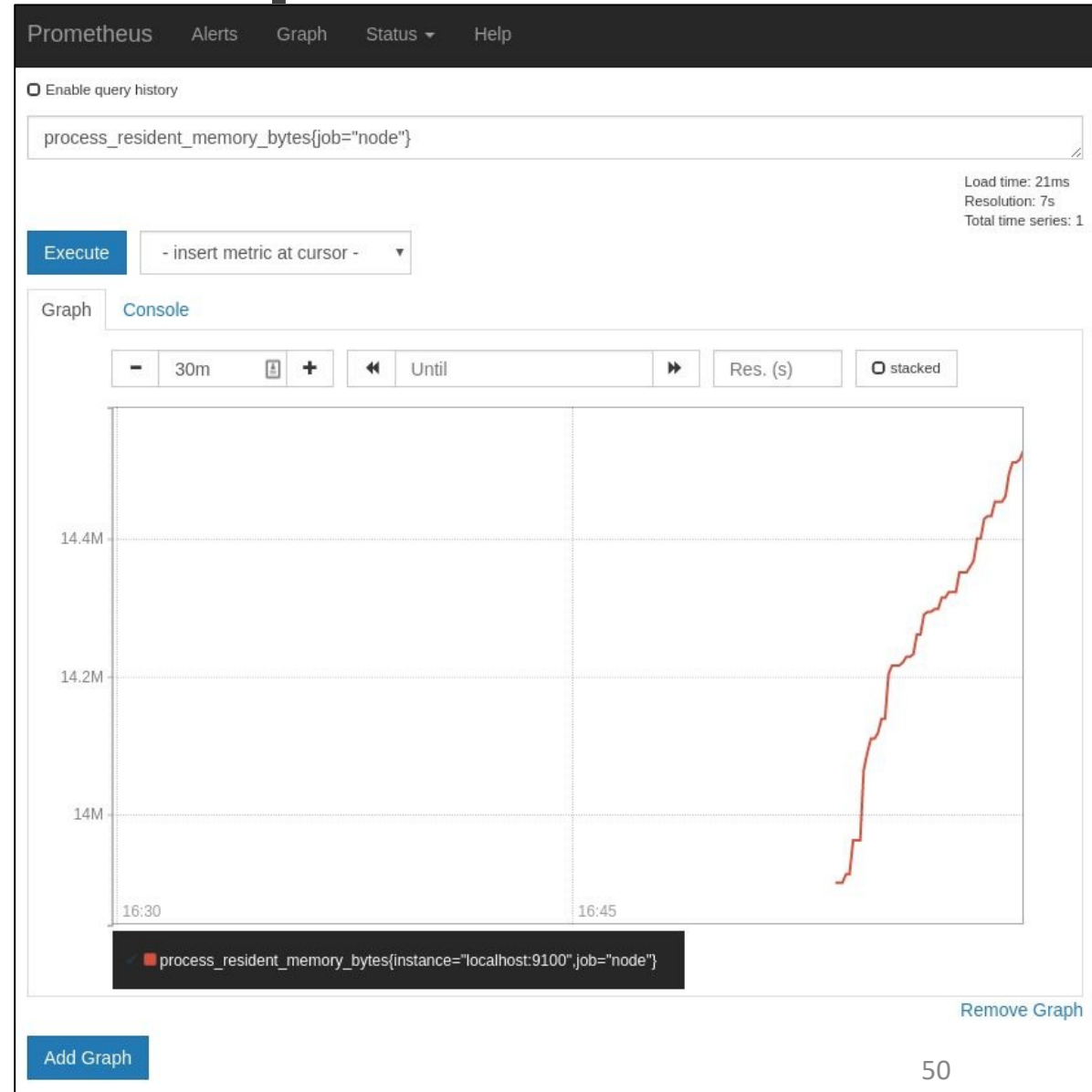
Add Graph

Executer le Node Exporter

4

Vous pouvez représenter graphiquement l'utilisation de la mémoire d'un seul nœud avec **`process_resident_memory_bytes{job="node"}`**.

Le `job = "node"` est appelé un selecteur d'étiquettes, et il restreint les métriques qui sont retournées



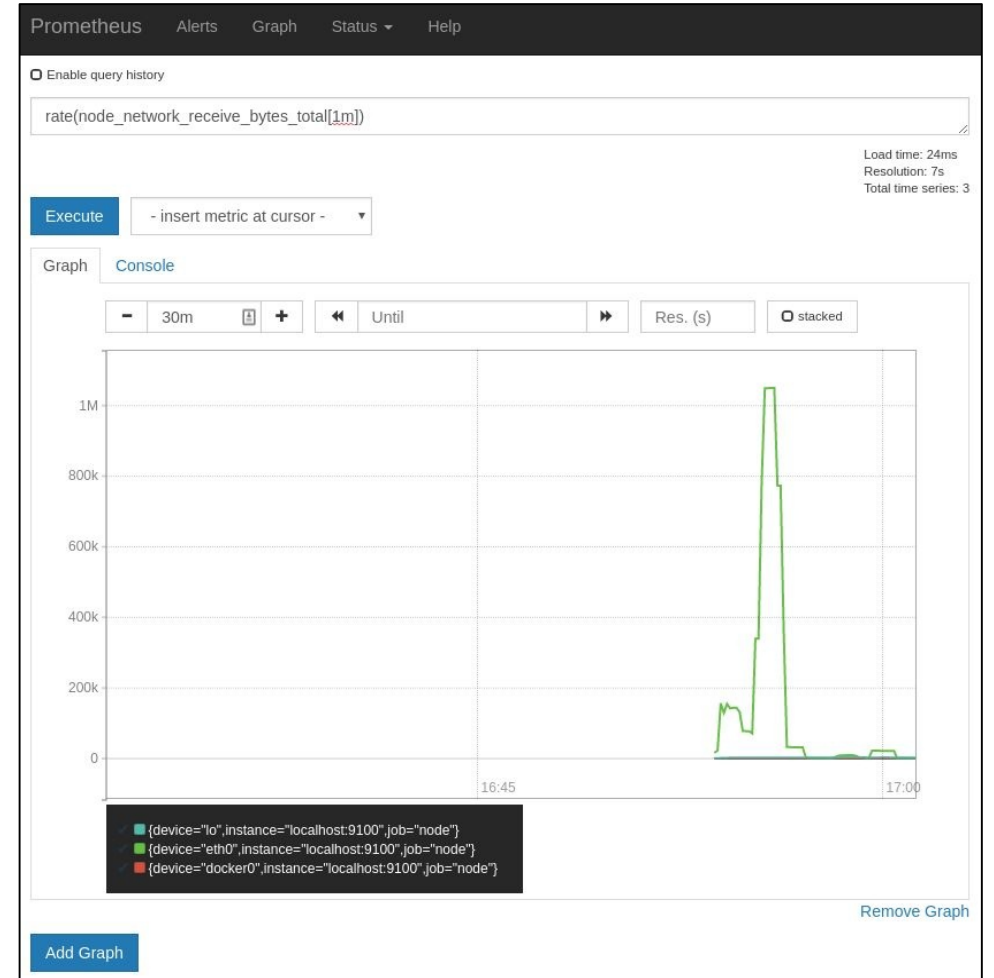
Executer le Node Exporter

5

Comme dernier exemple, évaluez le

`rate(node_network_receive_bytes_total[1m])`

en mode graphique



LAB 2 : Node Exporter

PromQL

Prometheus Query Language

Les types des métriques

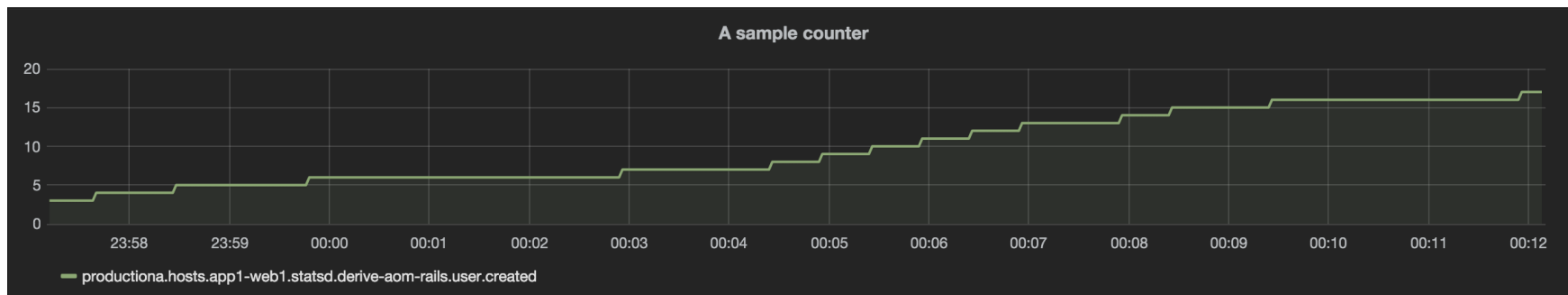
Il existe quatre types standard de métriques dans l'instrumentation Prometheus:

- **Counter (node_cpu_seconds_total)**
- **Gauge (node_memory_MemFree_bytes)**
- **Histogram (prometheus_http_request_duration_seconds)**
- **Summary (go_gc_duration_seconds)**

Les types des métriques

Compteur (Counter)

- Un compteur compte les choses.
- Les compteurs sont des nombres qui augmentent avec le temps et ne diminuent jamais.
- les compteurs peuvent parfois se remettre à zéro et recommencer à s'incrémenter.
- Exp: - nombre d'octets envoyés et reçus par un périphérique
- le nombre de connexions.
- Des exemples de compteurs commerciaux peuvent être le nombre de ventes par mois ou le nombre de commandes reçues par une application.



Les types des métriques

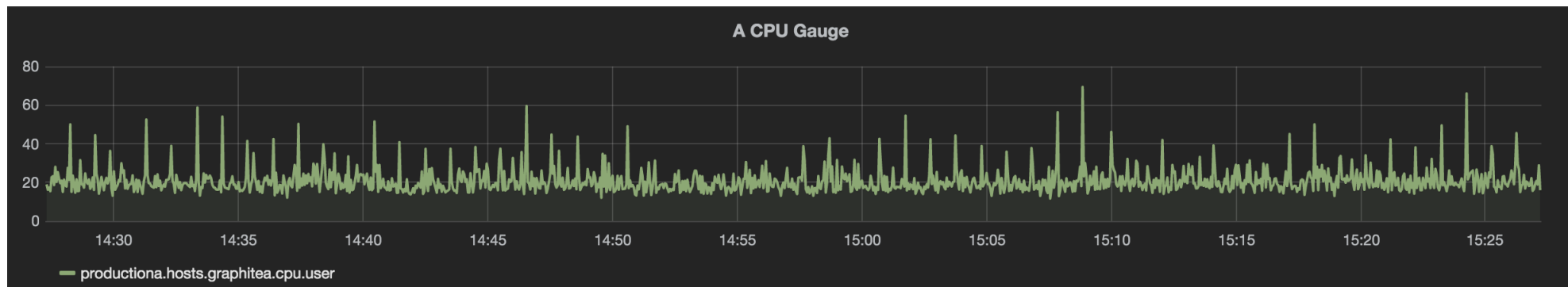
Compteur (Counter)

- Une chose utile à propos des compteurs est qu'ils vous permettent de calculer les taux de changement (rate).
- Chaque valeur observée est un instant dans le temps: t .
- Vous pouvez soustraire la valeur à t de la valeur à $t + 1$ pour obtenir le taux de changement entre les deux valeurs.
- Beaucoup d'informations utiles peuvent être comprises en comprenant le taux de changement entre deux valeurs.
- Par exemple, le nombre de connexions est intéressant, mais créer un taux à partir de celui-ci aide à identifier les périodes de popularité du site.

Les types des métriques

Jauge (Gauge)

- Les jauges sont des nombres qui devraient augmenter ou diminuer.
- Une jauge est essentiellement une instantanée d'une mesure spécifique.
- Les mesures classiques de l'utilisation du processeur, de la mémoire et du disque sont généralement articulées sous forme de jauges.
- Pour les mesures commerciales, une jauge peut être le nombre de clients présents sur un site.



Les types des métriques

Jauge (Gauge)

- Les jauges sont un instantané de l'état, et généralement lors de leur agrégation, vous voulez prendre une somme, une moyenne, un minimum ou un maximum.
- Considérez la métrique *node_filesystem_size_bytes* de votre Node Exporter, qui indique la taille de chacun de vos systèmes de fichiers montés et a des étiquettes de device, de fstype et de mountpoint.
- Vous pouvez calculer la taille totale du système de fichiers sur chaque machine avec:

```
sum without(device, fstype, mountpoint)(node_filesystem_size_bytes)
```

- Cela fonctionne comme *without* dit à l'agrégateur de sum de sommer tout avec les mêmes étiquettes, en ignorant ces trois.

Les types des métriques

① Jauge (Gauge)

- Vous pouvez utiliser la même approche avec d'autres agrégations.
- max vous indiquera la taille du plus grande des systèmes de fichiers montés sur chaque machine:

```
max without(device, fstype, mountpoint)(node_filesystem_size_bytes)
```


Les types des métriques

① Résumé (summary)

- Un résumé est une combinaison d'autres types, pour rendre les modèles communs plus simples à utiliser.
- Un résumé se compose de deux compteurs, et éventuellement de quelques jauges.
- Les métriques récapitulatives sont utilisées pour suivre la taille des événements, généralement leur durée, via la méthode **observe**.

Les types des métriques

Histogram

- Un histogramme est une métrique qui échantillonne les observations.
- Il s'agit d'une distribution de fréquence d'un ensemble de données.
- Vous regroupez les données - un processus appelé «regroupement» - et vous présentez les groupes de telle sorte que leurs tailles relatives soient visualisées.
- Chaque observation est comptée et placée dans des barres.
- Il en résulte plusieurs métriques: une pour chaque compartiment, plus des métriques pour la somme et le nombre de toutes les valeurs.

Types de données

Dans le langage d'expression de Prometheus, une expression ou une sous-expression peut être évaluée à l'un des quatre types suivants:

- **Scalaire** - une simple valeur numérique à virgule flottante
- **Vecteur instantané** - un ensemble de séries chronologiques contenant un seul échantillon pour chaque série chronologique, partageant tous le même horodatage
- **Vecteur de plage** - un ensemble de séries chronologiques contenant une plage de points de données au fil du temps pour chaque série chronologique
- **String** - une valeur de chaîne simple; actuellement inutilisé

Time series Selectors

Instant vector selectors (Sélecteurs de vecteurs instantanés)

- Permettent de sélectionner un ensemble de time series
- Une seule valeur d'échantillon pour chaque time series à un horodatage (instant) donné: le selecteur spécifie un seul nom de métrique.
- Il en résulte un vecteur instantané contenant des éléments pour toutes les séries temporelles qui portent ce nom de métrique.

Time series Selectors

Instant vector selectors (Sélecteurs de vecteurs instantanés)

Cet exemple sélectionne toutes les séries temporelles qui portent le nom **http_requests_total** de la métrique:

prometheus_http_requests_total

Time series Selectors

Instant vector selectors (Sélecteurs de vecteurs instantanés)

Il est possible de filtrer davantage ces séries chronologiques en ajoutant une liste séparée par des virgules d'étiquettes (Labels) entre accolades ({}).

Cet exemple sélectionne uniquement les séries temporelles avec le nom de la métrique *http_requests_total* qui ont également le libellé job défini comme prometheus et leur libellé group défini comme canary:

```
prometheus_http_requests_total{job="prometheus",group="canary"}
```

Time series Selectors

Instant vector selectors (Sélecteurs de vecteurs instantanés)

Il est également possible de faire correspondre inversement une valeur d'étiquette ou de faire correspondre les valeurs d'étiquette avec des expressions régulières.

Les opérateurs de correspondance d'étiquette suivants existent:

- `=` : Sélectionne des étiquettes qui sont exactement égales à la chaîne fournie.
- `!=` : Sélectionne des étiquettes qui ne sont pas égales à la chaîne fournie.
- `=~` : Sélectionne les libellés qui correspondent à la chaîne fournie.
- `!~` : Sélectionne les étiquettes qui ne correspondent pas à la chaîne fournie.

Time series Selectors

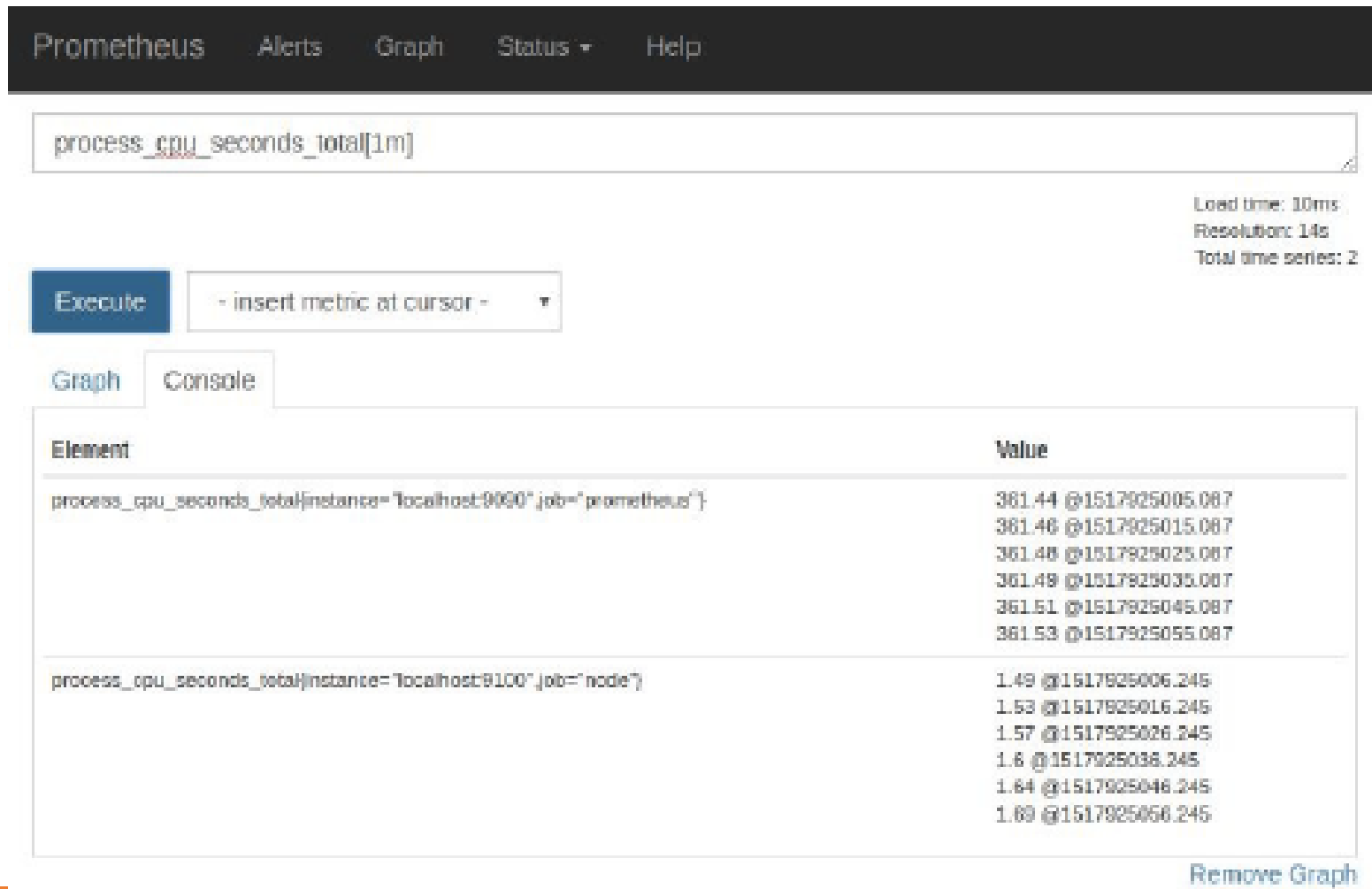
Range Vector selectors (Sélecteurs de vecteurs de plage)

- Le sélecteur de vecteurs de plage peut renvoyer de nombreux échantillons pour chaque série chronologique.
- Les vecteurs de plage sont toujours utilisés avec la fonction “rate”

`rate(process_cpu_seconds_total[1m])`

Time series Selectors

Range Vector selectors (Sélecteurs de vecteurs de plage)



Time series Selectors

Range Vector selectors (Sélecteurs de vecteurs de plage)

Durée →

Suffix	Meaning
ms	Milliseconds
s	Seconds, which have 1,000 milliseconds
m	Minutes, which have 60 seconds
h	Hours, which have 60 minutes
d	Days, which have 24 hours
w	Weeks, which have 7 days
y	Years, which have 365 days

Time series Selectors

Offset

- Il existe un modificateur que vous pouvez utiliser avec l'un ou l'autre type de sélecteur vectoriel appelé offset.
- Offset vous permet de prendre le temps d'évaluation d'une requête et de le replacer plus en arrière au niveau du sélecteur. Par exemple:

`process_resident_memory_bytes {job = "node"} offset 1h`

- Donne l'utilisation de la mémoire une heure avant l'instant d'évaluation de la requête.

Aggrégations

L'agrégation vous permet de résumer les métriques non seulement dans une seule application, mais aussi entre les applications.

Aggrégations

Grouping:

by

```
sum by(job, instance, device)(node_filesystem_size_bytes)
```

Without

```
sum without(fstype, mountpoint)(node_filesystem_size_bytes)
```

Aggrégations

Operators:

sum

```
sum without(fstype, mountpoint, device)(node_filesystem_size_bytes)
```

Count

```
count without(device)(node_disk_read_bytes_total)
```

Avg

```
avg without(cpu)(rate(node_cpu_seconds_total[5m]))
```

min / max

```
max without(device, fstype, mountpoint)(node_filesystem_size_bytes)
```

topk / bottomk

```
topk without(device, fstype, mountpoint)(2, node_filesystem_size_bytes)
```

quantile

```
quantile without(cpu)(0.9, rate(node_cpu_seconds_total{mode="system"}[5m]))
```

Opérateurs Binaires

Opérateurs arithmétiques + - * / % ^

Opérateurs logiques == != > < <= >=

Alerting

Alerting

L'alerting comporte deux parties.

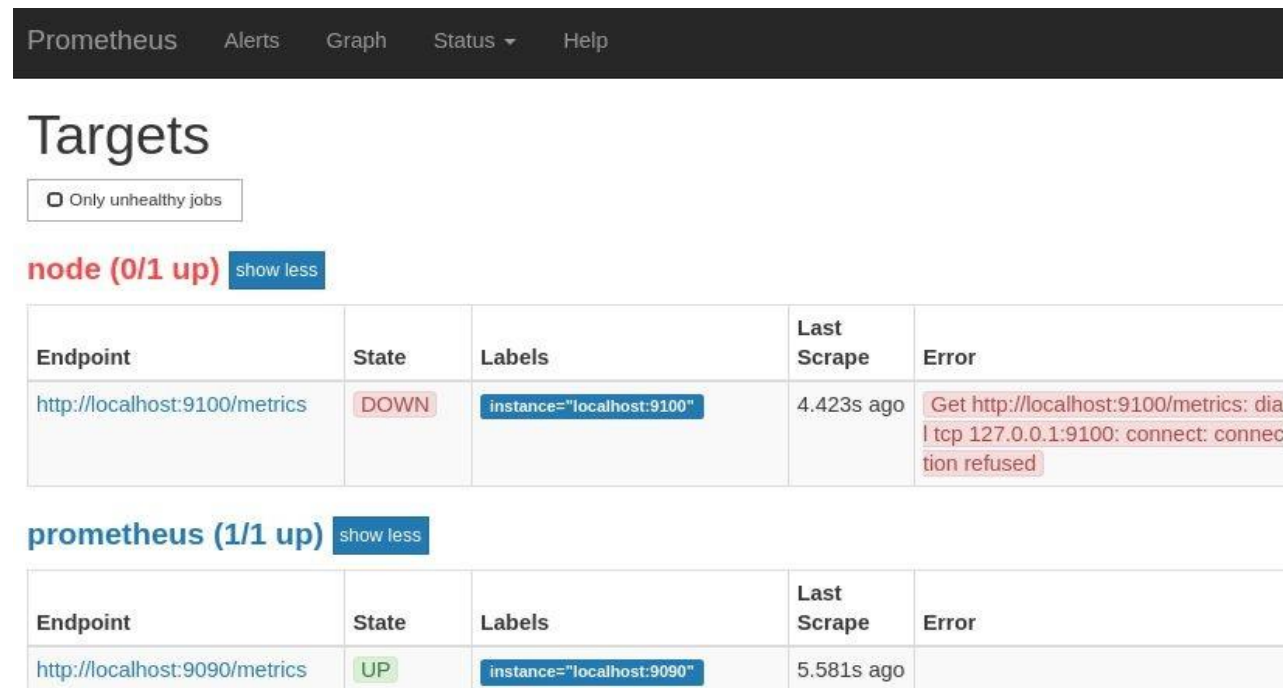
1. Ajout de règles d'alerte à Prometheus, définissant la logique de ce qui constitue une alerte.
2. Alertmanager convertit les alertes déclenchées en notifications, telles que des e-mails, des pages et des messages de discussion.

Alerting

Créer une condition sur laquelle vous voudrez peut-être alerter.

① Arrêtez l'exportateur de nœuds avec Ctrl-C.

Après le scrape suivant, la page **Targets** affichera le Node Exporter dans l'état **DOWN** avec l'erreur **connexion refusée** car rien n'écoute sur le port TCP et la requête HTTP est rejetée



The screenshot shows the Prometheus web interface. At the top is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this is the 'Targets' section, which includes a checkbox for 'Only unhealthy jobs'. There are two target groups: 'node (0/1 up)' and 'prometheus (1/1 up)'. Each group has a 'show less' button and a table of targets.

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9100/metrics	DOWN	instance="localhost:9100"	4.423s ago	Get http://localhost:9100/metrics: dial tcp 127.0.0.1:9100: connect: connection refused

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9090/metrics	UP	instance="localhost:9090"	5.581s ago	

Alerting

- ② La métrique **up** de l'Expression Browser a une valeur de 0 pour le Node Exporter

The screenshot shows the Prometheus Expression Browser interface. At the top is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below the navigation bar is a checkbox for 'Enable query history'. The main query input field contains the text 'up'. To the right of the input field, the following statistics are displayed: Load time: 17ms, Resolution: 14s, and Total time series: 2. Below the input field is an 'Execute' button and a dropdown menu with the text '- insert metric at cursor -'. Below the 'Execute' button are two tabs: 'Graph' and 'Console'. The 'Graph' tab is active, and it displays a table with two columns: 'Element' and 'Value'. The table contains two rows of data: one for 'up{instance="localhost:9090",job="prometheus"}' with a value of 1, and another for 'up{instance="localhost:9100",job="node"}' with a value of 0. Below the table is a 'Remove Graph' link. At the bottom left of the interface is an 'Add Graph' button.

Element	Value
up{instance="localhost:9090",job="prometheus"}	1
up{instance="localhost:9100",job="node"}	0

Alerting

2

Pour les règles d'alerte, vous avez besoin d'une expression **PromQL** qui renvoie uniquement les résultats sur lesquels vous souhaitez alerter.

Dans ce cas, c'est facile à le faire en utilisant l'opérateur **==**

The screenshot shows the Prometheus Alerts interface. At the top is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below the navigation bar is a toggle for 'Enable query history'. A text input field contains the query 'up == 0'. To the right of the input field, the following statistics are displayed: Load time: 11ms, Resolution: 14s, and Total time series: 1. Below the input field is an 'Execute' button and a dropdown menu with the option '- insert metric at cursor -'. Below these are two tabs: 'Graph' and 'Console'. The 'Console' tab is active, showing a table with two columns: 'Element' and 'Value'. The table contains one row with the element 'up{instance="localhost:9100",job="node"}' and the value '0'. At the bottom right of the table is a 'Remove Graph' link. At the bottom left is an 'Add Graph' button.

Element	Value
up{instance="localhost:9100",job="node"}	0

Alerting

③

On doit ajouter cette expression dans une règle d'alerte dans Prometheus:
1- Indiquer le **Alertmanager** et le fichier des règles d'alerte à utiliser, dans le fichier prometheus.yml

```
global:
  scrape_interval: 10s
  evaluation_interval: 10s
  rule_files:
    - rules.yml
  alerting:
    alertmanagers:
      - static_configs:
        - targets:
          - localhost:9093
```

...

Alerting

- ③ 2- créez un nouveau fichier **rules.yml** avec le contenu ci-dessous, puis redémarrez Prometheus.

```
groups:  
  - name: example  
    rules:  
      - alert: InstanceDown  
        expr: up == 0  
        for: 1m
```

Alerting

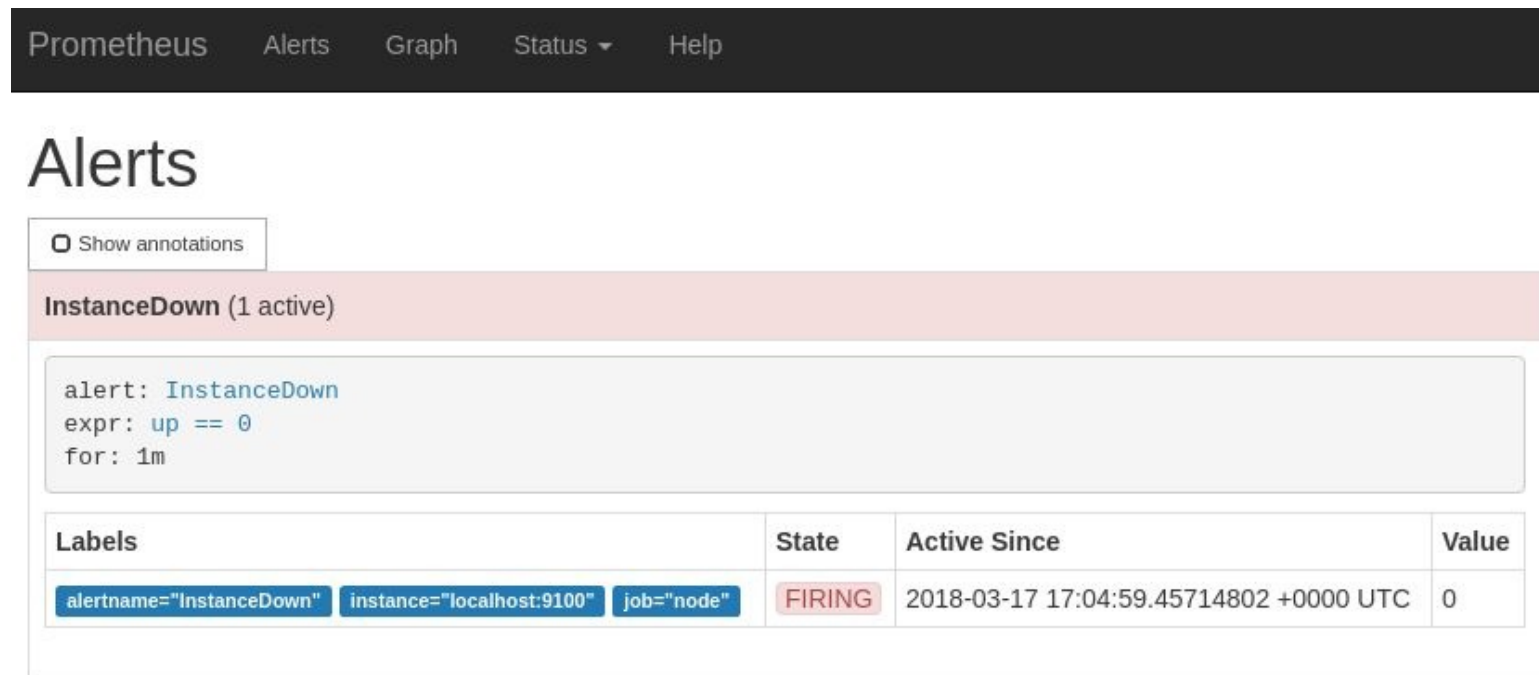
4

L'alerte **InstanceDown** sera évaluée toutes les 15 secondes conformément à **evaluation_interval**.

Si une série est renvoyée en continu pendant au moins une minute (le **for**), l'alerte sera alors considérée comme étant déclenchée (**firing**).

Jusqu'à ce que la minute requise soit écoulée, l'alerte sera en attente (**pending**).

Sur la page **Alerts**, vous pouvez cliquer sur cette alerte et voir plus de détails, y compris ses **labels**



The screenshot shows the Prometheus Alerts interface. At the top is a navigation bar with links for Prometheus, Alerts, Graph, Status, and Help. Below this is the 'Alerts' section header. A toggle switch for 'Show annotations' is visible. The main content area shows an 'InstanceDown' alert with 1 active instance. The alert details include the name 'InstanceDown', the expression 'up == 0', and the duration 'for: 1m'. Below the details is a table with columns for Labels, State, Active Since, and Value. The table contains one row for the active alert, which is in the 'FIRING' state and has been active since 2018-03-17 17:04:59.45714802 +0000 UTC, with a value of 0.

Labels	State	Active Since	Value
alertname="InstanceDown" instance="localhost:9100" job="node"	FIRING	2018-03-17 17:04:59.45714802 +0000 UTC	0

Alerting

- ⑤ Maintenant que vous avez une alerte déclenchée, vous avez besoin d'un **Alertmanager** pour en faire quelque chose.

Depuis <https://prometheus.io/download/>, téléchargez la dernière version d'Alertmanager

Décompressez **Alertmanager** et cd dans son répertoire.

```
hostname $ tar -xzf alertmanager-*.linux-amd64.tar.gz
hostname $ cd alertmanager-*.linux-amd64/
```


Alerting

- ⑥ Il existe plusieurs façons dont Alertmanager peut vous avertir, mais la plupart de celles prédéfinies utilisent des fournisseurs commerciaux et ont des instructions de configuration qui ont tendance à changer au fil du temps.
- Exemples : Email, pagerduty, slack, pushover, opsgenie, victorops, wechat
 - En cas de l'utilisation d'un receiver non défini par Alertmanager, on doit utiliser la configuration de webhook

Par conséquent, on va essayer d'envoyer des alertes par **email**, **Pagerduty** et par **slack**.



PagerDuty



slack

Alerting: Routing Tree

- ⑥ Le champ ***route*** spécifie la route de niveau supérieur, de secours ou par défaut.

Les itinéraires forment un arbre, donc vous pouvez et aurez généralement plusieurs itinéraires en dessous.

Par exemple :

```
route:
  receiver: fallback-pager
  routes:
    - match:
        severity: page
        receiver: team-pager
    - match:
        severity: ticket
        receiver: team-ticket
```

Alerting

a- Alert par email:

- ⑥ Vous devez baser votre **alertmanager.yml** comme ci-dessous, en ajustant **smtp_smarthost**, **smtp_from** et en fonction de votre configuration et de votre adresse e-mail.

```
global:
  smtp_smarthost: 'localhost:25'
  smtp_from: 'youraddress@example.org'
  smtp_auth_username: username
  smtp_auth_password: password
route:
  receiver: example-email
receivers:
  - name: example-email
    email_configs:
      - to: 'youraddress@example.org'
```

Alerting

a- Alert par email: GMail

6

myaccount.google.com/u/2/security

Google Account Search Google Account

Security

Settings and recommendations to help you keep your account secure

- Home
- Personal info
- Data & personalization
- Security**
- People & sharing
- Payments & subscriptions

Security issues found

Protect your account now by resolving these issues

[Secure account](#)

Signing in to Google

Password	Last changed 7:54 PM	>
2-Step Verification	On	>
App passwords	None	>

Alerting

6

b- Alert par Pagerduty:

Créer une application Slack

Sur Pagerduty: Configuration -> Services -> + New Service

Puis dans l'onglet Integration, copier le Token

Configurer AlertManager

```
global:
  pagerduty_url: 'https://events.pagerduty.com/v2/enqueue'
route:
  group_by: ['instance', 'alert']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 3h
  receiver: 'pagerduty-notifications'
receivers:
- name: 'pagerduty-notifications'
  pagerduty_configs:
  - service_key: 0c1cc665a594419b6d215e81f4e38f7
    send_resolved: true
```

Alerting

c- Alert par slack:

6

Créer une application Slack

<https://blog.ruanbekker.com/blog/2019/04/18/setup-a-slack-webhook-for-sending-messages-from-applications/>

Configurer AlertManager

```
global:
  slack_api_url: 'https://hooks.slack.com/services/x/xx/xxx'
route:
  group_by: ['instance', 'alert']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 3h
  receiver: team-1
receivers:
  - name: 'team-1'
    slack_configs:
      # https://prometheus.io/docs/alerting/configuration/#slack config
      - channel: 'system_events'
      - username: 'AlertManager'
      - icon_emoji: ':joy:'
```

Alerting: Labels

- ⑥ • Les Labels des alertes définissent leurs identités
- Dans la route, on ajoute un des labels pour filtrer les métriques à considérer par l'alerte
- Exemple:
 - alert: InstanceDown
 - expr: up{job="node"} == 0
 - for: 1h
 - labels:
 - severity: ticket

Annotations : Notification templates

⑥

- Le champ d'annotations vous permet de fournir des informations supplémentaires sur une alerte, comme une brève description de ce qui ne va pas.
- Ecrites avec Go template syntax (<https://golang.org/pkg/text/template/>)
- Possibilité d'exposer certaines variables qui contiennent les Labels et la valeur d'une time-series en utilisant **\$labels** et **\$value**

Exemple

annotations:

summary: "Et zut, on arrive a moins de 20Go sur {{ \$labels.instance }}"

description: "Il reste precisement {{ \$value }}"

Grouping Alerts

- ⑥ • Alertmanager met toutes les alertes pour une route dans un seul groupe, ce qui signifie que vous recevrez une grosse notification.
- Bien que cela puisse convenir dans certains cas, vous souhaiterez généralement que vos notifications soient un peu plus petites.
- Le champ `group_by` vous permet de spécifier une liste d'étiquettes par lesquelles regrouper les alertes;

Silencing Alert

- ⑥ • Si vous connaissez déjà un problème ou si vous interrompez un service pour maintenance, il est inutile de continuer l'appel à ce sujet.
- Les silences vous permettent d'ignorer certaines alertes pendant un certain temps et sont ajoutés via l'Interface Web d'Alertmanager.

Dashboarding with Grafana

Qu'est ce que Grafana ?

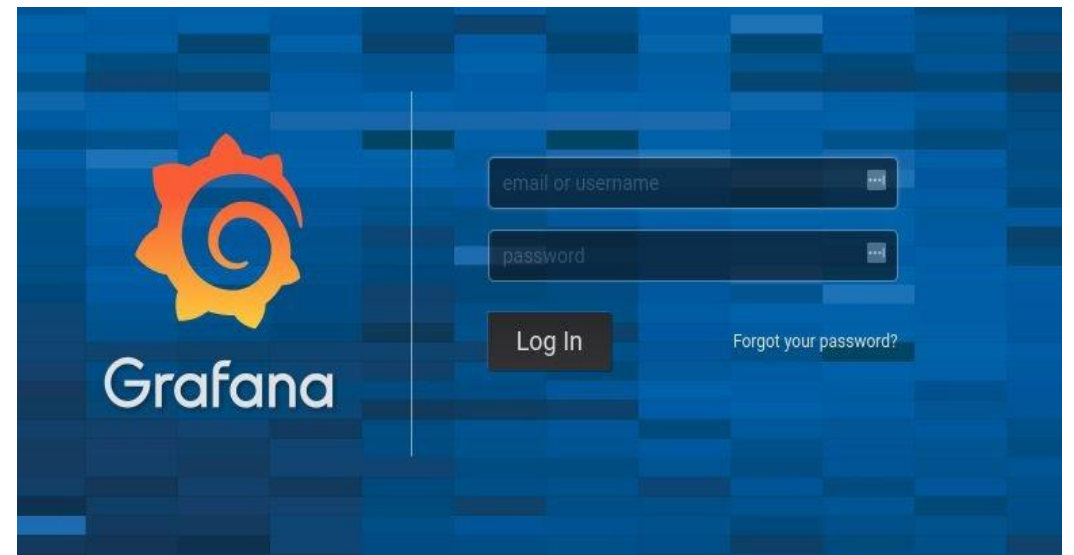
- Grafana est un outil populaire avec lequel vous pouvez créer des tableaux de bord pour de nombreux systèmes de surveillance et de non-surveillance différents, notamment Graphite, InfluxDB, Elasticsearch et PostgreSQL.
- Il s'agit de l'outil recommandé pour créer des tableaux de bord lors de l'utilisation de Prometheus et améliore continuellement son support à Prometheus.

Installation

- ① Vous pouvez télécharger Grafana sur <https://grafana.com/grafana/download>. Le site comprend des instructions d'installation, mais si vous utilisez Docker, par exemple, vous tapez:

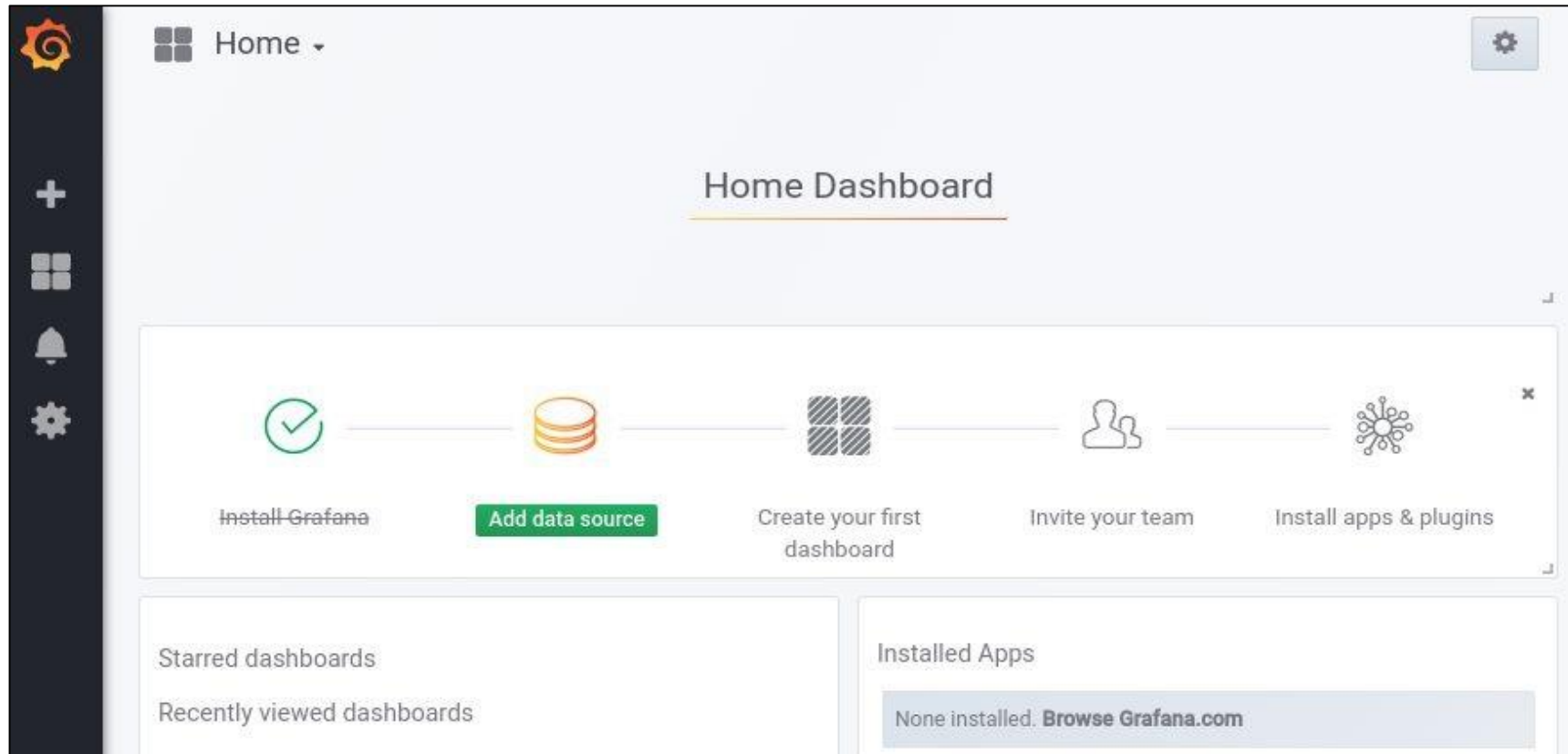
```
docker run -d --name=grafana --net=host grafana/grafana:5.0.0
```

Une fois Grafana installé, vous devriez pouvoir y accéder dans votre navigateur à **[http: //localhost:3000/](http://localhost:3000/)**, et vous verrez un écran de connexion



Installation

- ② Connectez-vous avec le nom d'utilisateur par défaut **admin** et le mot de passe par défaut, qui est également **admin**. Vous devriez voir le tableau de bord d'accueil



Data Source

- Grafana utilise des sources de données pour récupérer les informations illustrées dans les graphiques.
- Il existe une variété de types de sources de données prises en charge, y compris OpenTSDB, PostgreSQL et, bien sûr, Prometheus.
- Vous pouvez avoir plusieurs sources de données du même type, et généralement vous en avez une par Prometheus en cours d'exécution.
- Un tableau de bord Grafana peut avoir des graphiques provenant de diverses sources, et vous pouvez même mélanger des sources dans un dashboard.

Data Source

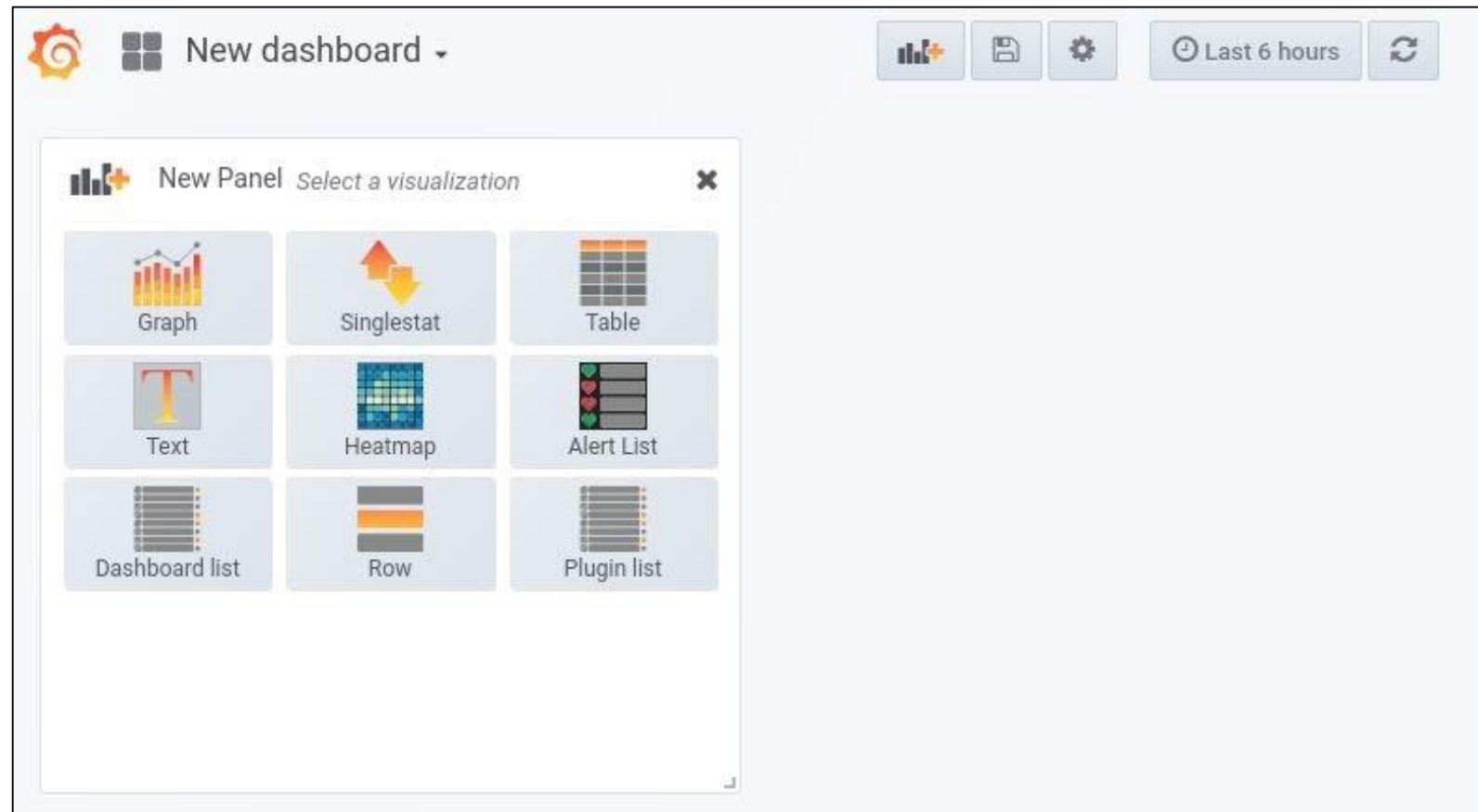
- Les versions plus récentes de **Grafana** facilitent l'ajout de votre première source de données.
- Cliquez sur Ajouter une source de données et ajoutez une source de données avec le nom de Prometheus, le type de Prometheus et une URL `http://localhost:9090`.
- Laissez tous les autres paramètres à leurs valeurs par défaut, puis cliquez sur Enregistrer et tester.
- Si cela fonctionne, vous recevrez un message indiquant que la source de données fonctionne.
- Si ce n'est pas le cas, vérifiez que le Prometheus fonctionne bien et qu'il est accessible depuis Grafana.

Data Source

- Les versions plus récentes de **Grafana** facilitent l'ajout de votre première source de données.
- Cliquez sur Ajouter une source de données et ajoutez une source de données avec le nom de Prometheus, le type de Prometheus et une URL `http://localhost:9090`.
- Laissez tous les autres paramètres à leurs valeurs par défaut, puis cliquez sur Enregistrer et tester.
- Si cela fonctionne, vous recevrez un message indiquant que la source de données fonctionne.
- Si ce n'est pas le cas, vérifiez que le Prometheus fonctionne bien et qu'il est accessible depuis Grafana.

Dashboard & Panneaux

- Revenez à <http://localhost:3000/> dans votre navigateur, et cette fois cliquez sur Nouveau tableau de bord,



Dashboard & Panneaux

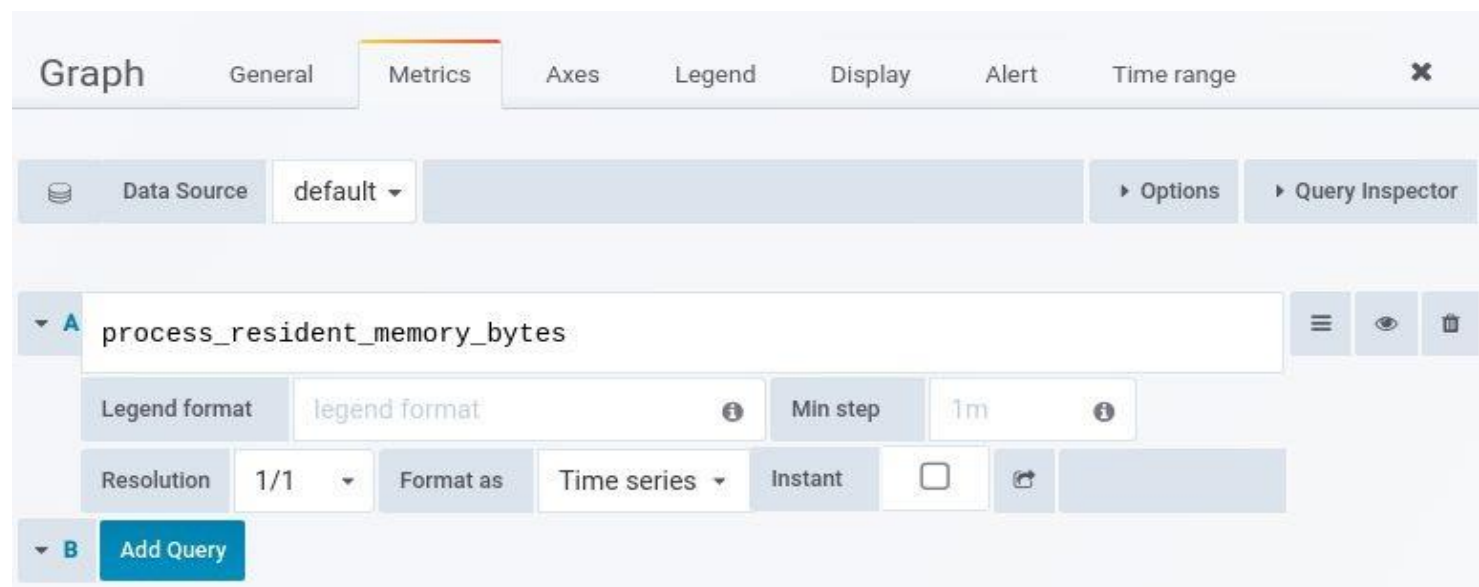
- À partir de là, vous pouvez sélectionner le premier panneau que vous souhaitez ajouter.
- Les panneaux sont des zones rectangulaires contenant un graphique, un tableau ou d'autres informations visuelles.
- Vous pouvez ajouter de nouveaux panneaux au-delà du premier avec le bouton Ajouter un panneau, qui est le bouton sur la ligne supérieure avec le signe plus orange.
- Depuis Grafana 5.0.0, les panneaux sont organisés dans un système de grille et peuvent être réorganisés par glisser-déposer.

Graph Panel

- Le **Graph Panel** est le panneau principal que vous utiliserez.
- Comme son nom l'indique, il affiche un graphique.
- Cliquez sur le bouton **Graph** pour ajouter un panneau graphique.
- Vous avez maintenant un graphique vierge.
- Pour le configurer, cliquez sur **Panel Title** puis sur **Edit**.

Graph Panel

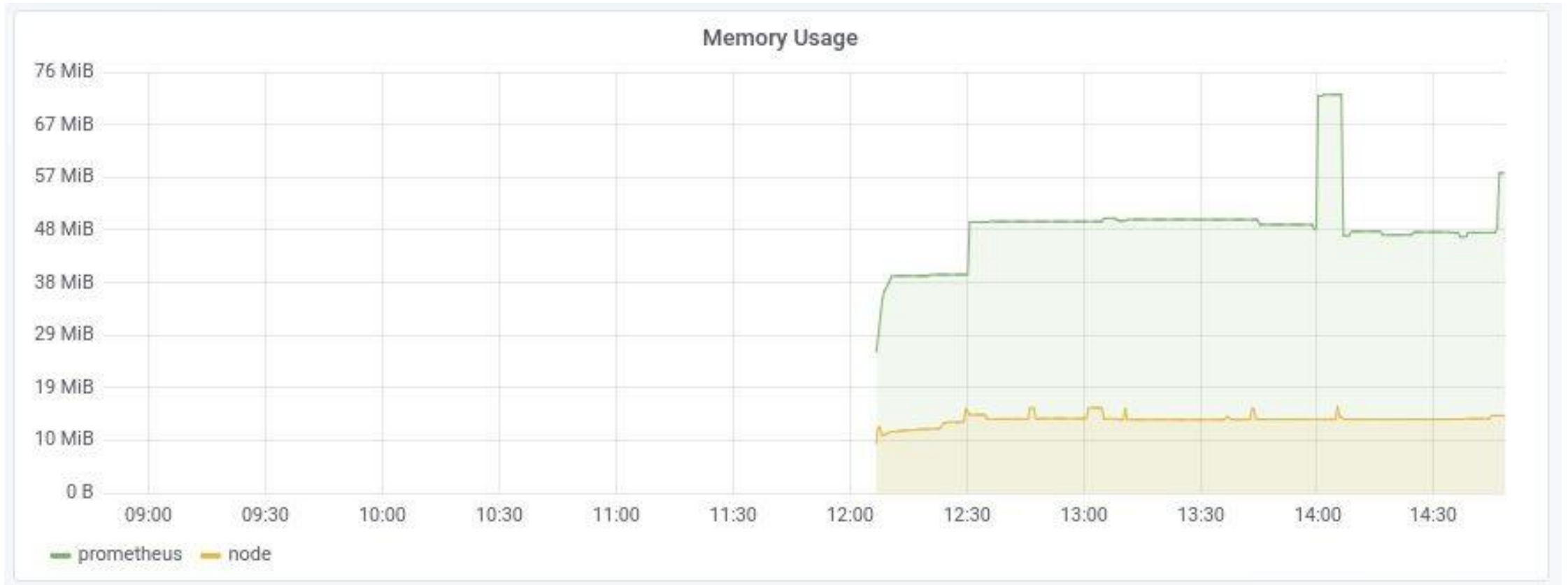
- L'éditeur de graphiques s'ouvrira sur l'onglet Métriques.
- Entrez **process_resident_memory_bytes** pour l'expression de requête, dans la zone de texte à côté de A, puis cliquez à l'extérieur de la zone de texte.
- Vous verrez un graphique de l'utilisation de la mémoire la même expression a été utilisée dans le navigateur d'expression.



Graph Panel

- Grafana offre plus que le navigateur d'expression (Expression Browser).
- Vous pouvez configurer la légende pour afficher autre chose que le nom de time-series.
- Mettez **{{job}}** dans la zone de texte Legend Format.
- Sous l'onglet Axes, modifiez l'unité Y gauche en données/octetets.
- Sous l'onglet Général, modifiez le titre en utilisation de la mémoire.

Graph Panel




Time Controls

- Vous avez peut-être remarqué les time controls de Grafana en haut à droite de la page.
- Par défaut, il doit indiquer «6 dernières heures».
- Cliquez sur les time controls choisir une plage de temps et la fréquence de rafraîchissement.
- Les contrôles de temps s'appliquent à un tableau de bord entier à la fois, mais vous pouvez également configurer certains remplacements par panneau.

Time Controls


 New dashboard





 Last 6 hours



Custom range

From:


To:


Refreshing every:


Apply

Quick ranges

Last 2 days	Yesterday	Today	Last 5 minutes
Last 7 days	Day before yesterday	Today so far	Last 15 minutes
Last 30 days	This day last week	This week	Last 30 minutes
Last 90 days	Previous week	This week so far	Last 1 hour
Last 6 months	Previous month	This month	Last 3 hours
Last 1 year	Previous year	This month so far	<u>Last 6 hours</u>
Last 2 years		This year	Last 12 hours
Last 5 years		This year so far	Last 24 hours

Singlestat Panel

- Le panneau Singlestat affiche la valeur d'une seule time-series.
- Ajoutons une valeur de série chronologique.
- Par défaut, le panneau Singlestat calculera la moyenne de la série chronologique sur la plage de temps du tableau de bord.
- Ce n'est souvent pas ce que vous voulez, donc dans l'onglet Options, changez Stat en Current.
- Le texte par défaut peut être un peu petit, modifiez donc la taille de police à 200%.
- Sous l'onglet Général, remplacez le titre par Prometheus Time Series.
- Enfin, cliquez sur Retour au tableau de bord

Singlestat Panel

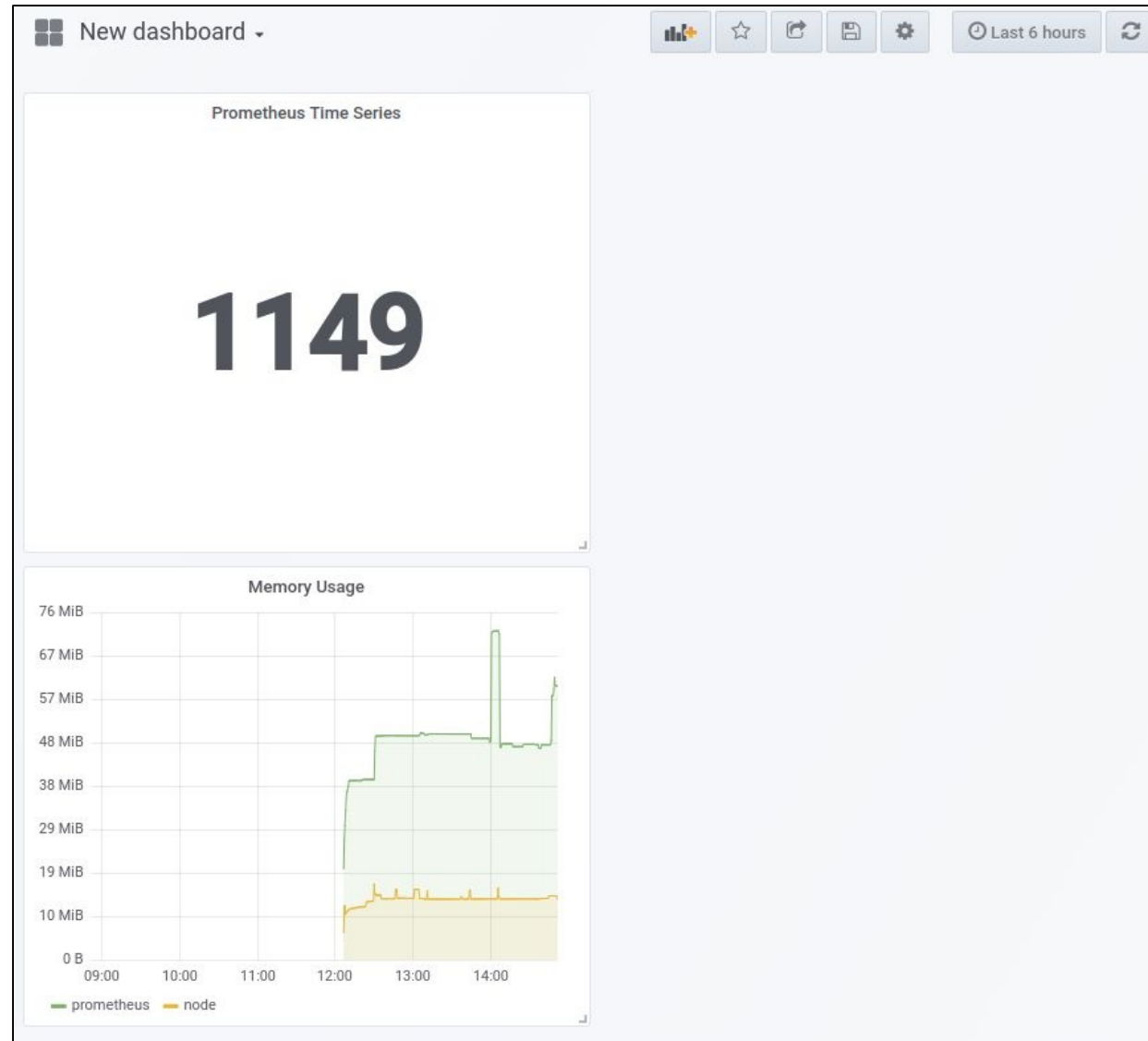


Table Panel

- Permet d'afficher des time series multiples
- Ajouter un panneau pour l'expression `rate(node_network_receive_bytes_total[1m])`
- Cocher Instant checkbox.
- modifier la règle de temps existante pour avoir un type de Hidden
- Cliquez sur + Add pour ajouter une nouvelle règle avec Apply to colonnes nommées job avec Type de Hidden
- ajoutez une autre règle masquant l'instance. Pour définir l'unité, + Add une règle pour la colonne Valuer et définissez son unité sur Bytes/sec sous le débit de données.
- Enfin, sous l'onglet Général, définissez le titre sur Trafic réseau reçu



THANK YOU