

Mode Opérateur

Importer et maintenir les dashboards Grafana

Code : MO-PLT-024
Version : 1.1
Date : 26 juin 2026
Auteur : Cédric LEGRAND
Classification : USAGE INTERNE — Équipe BTS SIO

Historique des révisions

Version	Date	Modifications
1.0	13/06/2026	Création initiale — méthode de vérification de compatibilité
1.1	26/06/2026	Harmonisation de la présentation des étapes et des encadrés.

1 Objet

Ce mode opératoire décrit l'ajout et la maintenance des **dashboards Grafana** de la supervision BTS SIO : choisir un dashboard communautaire sur grafana.com, **vérifier sa compatibilité avant import**, l'importer hors-ligne par l'API, corriger une requête cassée et organiser l'ensemble (dossier, page d'accueil).

L'étape de vérification est le cœur du document. Un dashboard référence des noms de métriques figés au moment de sa publication ; or les exporteurs renomment des métriques au fil de leurs versions majeures. Le symptôme est trompeur : la collecte fonctionne, la cible est UP, mais les panneaux affichent « N/A » ou « No data ». Le cas s'est présenté sur cette infrastructure : le dashboard Windows historique (ID 14694) interrogeait les métriques `windows_cs_*`, supprimées par `windows_exporter` 0.31 — près de la moitié de ses requêtes ne correspondait plus à rien.

2 Champ d'application

Public concerné	Administrateurs de l'infrastructure BTS SIO
Application	Grafana OSS 13.0.1 — CT 200 <code>docker-srv</code> , https://grafana.docker.bts.sio
Source de données	Prometheus (datasource par défaut)
Compte	<code>admin</code> (coffre Vaultwarden, item « Grafana »)
Outils	navigateur, <code>curl</code> , <code>grep</code> , Python 3 sur le poste d'administration
Durée	20 à 30 minutes pour un import vérifié

3 Concepts

3.1 Les dashboards en service

Dashboard	Usage	Origine
Vue d'ensemble — Infrastructure BTS SIO	Page d'accueil : état des cibles, jauges CPU/RAM/disque par hôte, tendances, conteneurs, alertes actives	local (bts-overview)
Windows Exporter Dashboard 2025	Détail DC1/DC2 (compatible windows_exporter 0.31+)	grafana.com #23942
cAdvisor exporter — Docker containers Overview	Détail des conteneurs Docker du CT 200	grafana.com #21743
Node Exporter Full	Détail des serveurs Linux supervisés	grafana.com #1860

Tous sont rangés dans le dossier **Infrastructure BTS SIO** ; la vue d'ensemble est définie comme tableau de bord d'accueil de l'organisation.

3.2 Liaison à la source de données

Un dashboard communautaire ne connaît pas la source de données locale : son JSON contient une variable d'entrée (`DS_PROMETHEUS`) à lier au moment de l'import. En passant par l'API, cette liaison se fait avec l'**UID** de la datasource ; en passant par l'interface graphique, Grafana propose un menu déroulant équivalent.

3.3 Pourquoi un dashboard affiche « N/A »

Trois causes, par ordre de fréquence :

1. **Métriques renommées** : le dashboard vise une autre génération d'exporteur que celle déployée — c'est la cause la plus sournoise, traitée par la vérification de compatibilité ci-dessous ;
2. **datasource non liée** : l'import a été fait sans renseigner la variable d'entrée ;
3. **variables de gabarit vides** : les menus déroulants (`job`, `instance`) sont alimentés par une requête `label_values()` qui porte elle-même sur une métrique absente — tout le dashboard paraît alors vide.



Contrainte hors-ligne

Le CT 200 n'a pas d'accès Internet : le bouton « Import via grafana.com » de l'interface ne fonctionne pas. Le JSON est téléchargé sur le poste d'administration, puis poussé par l'API ou collé dans l'interface.

4 Prérequis



Prérequis

- Poste d'administration connecté au VPN WireGuard et à Internet
- Identifiants admin de Grafana (coffre Vaultwarden, item « Grafana »)
- Accès SSH au CT 200 pour les appels d'API en local (`127.0.0.1:3000`)
- **UID de la datasource Prometheus** : `curl -su admin:*** http://127.0.0.1:3000/api/datasources (champ uid)`

5 Procédure

5.1 Choisir et vérifier un dashboard

1 Identifier un candidat sur grafana.com

Sur <https://grafana.com/grafana/dashboards/>, filtrer par source de données *Prometheus* et rechercher le nom de l'exporteur. Trois critères de tri :

- date de mise à jour : un dashboard maintenu suit les renommages de métriques ;
- version d'exporteur annoncée : certains titres la mentionnent explicitement (« v0.31+ compatible ») ;
- popularité : un grand nombre de téléchargements ne garantit pas la compatibilité — les dashboards les plus téléchargés sont souvent les plus anciens.

Télécharger le JSON de la dernière révision :

```
curl -sL -o dashboard.json \  
  "https://grafana.com/api/dashboards/23942/revisions/latest/download"
```

2 Vérifier la compatibilité des métriques avant import

Le principe : extraire les noms de métriques référencés par le dashboard, puis les confronter à la liste des métriques réellement présentes dans Prometheus. Depuis le CT 200 :

```
# 1. Métriques disponibles dans Prometheus  
curl -s "http://127.0.0.1:9090/api/v1/label/__name__/values" \  
  | python3 -c "import json,sys; \  
    print('\n'.join(json.load(sys.stdin)['data']))" \  
  > metriques_disponibles.txt
```

```
# 2. Métriques référencées par le dashboard
grep -oE '(windows|node|container)_[a-z0-9_]+' dashboard.json \
| sort -u > metriques_dashboard.txt

# 3. Croisement : ce qui manque
comm -23 metriques_dashboard.txt \
<(sort metriques_disponibles.txt)
```



```
root@docker-srv: ~
root@docker-srv:~# curl -s 'http://127.0.0.1:9090/api/v1/label/__name__/values' \
| python3 -c "import json,sys; \
print('\n'.join(json.load(sys.stdin)['data']))" \
> metriques_disponibles.txt
grep -oE '(windows|node|container)_[a-z0-9_]+' dashboard.json \
| sort -u > metriques_dashboard.txt
comm -23 metriques_dashboard.txt <(sort metriques_disponibles.txt)
windows_exporter
windows_system_system_up_time
```

Figure 1 : Croisement des métriques référencées par le dashboard avec celles présentes dans Prometheus : une seule métrique manquante, facile à corriger ; un dashboard incompatible en afficherait des dizaines.



Interpréter le résultat

Une liste vide ou presque : le dashboard est compatible. Une ou deux métriques manquantes : importable, puis corriger les panneaux concernés (étape 5). Une longue liste — typiquement des familles entières comme `windows_cs_*` — : chercher un dashboard d'une génération plus récente plutôt que de rapiécer celui-ci.

5.2 Importer

3

Importer par l'API en liant la datasource

Copier le JSON sur le CT 200 (scp), puis l'importer dans le dossier cible en liant la variable d'entrée à l'UID de la datasource :

```
python3 - <<'EOF'
import json, urllib.request, base64

dash = json.load(open("/tmp/dashboard.json"))
payload = {
    "dashboard": dash,
    "overwrite": True,
    "folderUid": "infra-bts",
    "inputs": [{"name": "DS_PROMETHEUS",
                  "type": "datasource",
                  "pluginId": "prometheus",
                  "value": "cfiayvlqqhwcgc"}]    # UID datasource
}
req = urllib.request.Request(
    "http://127.0.0.1:3000/api/dashboards/import",
    data=json.dumps(payload).encode(),
    headers={"Content-Type": "application/json",
             "Authorization": "Basic " +
                 base64.b64encode(b"admin:MOT_DE_PASSE").decode()})
print(json.load(urllib.request.urlopen(req)))
EOF
```

La réponse contient `importedUrl`, le chemin du dashboard créé.



```
root@docker-srv: ~  
root@docker-srv:~# python3 - <<'EOF'  
import json, urllib.request, base64  
dash = json.load(open("/tmp/dashboard.json"))  
payload = {"dashboard": dash, "overwrite": True,  
          "folderUid": "infra-bts",  
          "inputs": [{"name": "DS_PROMETHEUS", "type": "datasource",  
                    "pluginId": "prometheus", "value": "cfiayvlqhwgcg"}]}  
req = urllib.request.Request(  
    "http://127.0.0.1:3000/api/dashboards/import", ...)  
print(json.load(urllib.request.urlopen(req)))  
EOF  
{'title': 'Windows Exporter Dashboard 2025 (v0.31+ compatible)', 'imported': True, 'importedUrl':  
  '/d/Kdh00oSGz/windows-exporter-dashboard-2025-v0-312b-compatible'}
```

Figure 2 : Import du dashboard par l'API avec liaison de la datasource : la réponse retourne le titre et l'URL du dashboard importé.



Variante graphique

Le même import se fait sans API : menu **Dashboards** → **New** → **Import**, coller le contenu du JSON, choisir le dossier et la datasource dans les menus, puis **Import**. L'API reste préférable pour la reproductibilité et pour consigner l'opération.

5.3 Corriger et organiser

4

Corriger une requête incompatible

Pour une métrique manquante isolée, corriger le panneau concerné : l'ouvrir (Edit), remplacer l'ancien nom par l'équivalent actuel dans le champ de requête, puis **Save dashboard**. Exemple réel rencontré lors de l'import du dashboard Windows 2025 :

```
# Panneau ñ Uptime ž métrique supprimée en 0.31 :  
time() - windows_system_system_up_time{...}  
# Équivalent actuel (même sémantique, timestamp de boot) :  
time() - windows_system_boot_time_timestamp{...}
```

La correspondance ancien → nouveau nom se trouve dans le journal des changements (*CHANGELOG*) de l'exporteur concerné. Pour corriger avant import, le remplacement peut aussi se faire directement dans le JSON (*sed*).

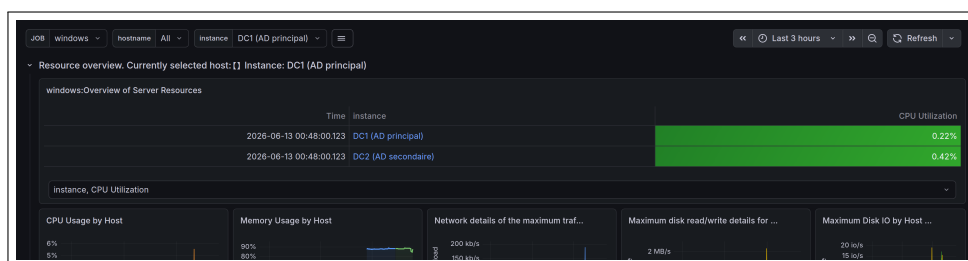


Figure 3 : Panneau corrigé après remplacement de la métrique supprimée : l'uptime des deux contrôleurs de domaine s'affiche.

5

Ranger et définir la page d'accueil

Conserver tous les dashboards d'infrastructure dans le dossier Infrastructure BTS SIO. Pour changer le tableau de bord d'accueil de l'organisation : Administration → Default preferences → *Home Dashboard*, ou par l'API :

```
curl -X PUT http://127.0.0.1:3000/api/org/preferences \
  -H "Content-Type: application/json" \
  -u admin:MOT_DE_PASSE \
  -d '{"homeDashboardUID": "bts-overview"}'
```



Figure 4 : La vue d'ensemble en page d'accueil : état des huit cibles, jauges par hôte et alertes actives, visibles dès la connexion.

6 Vérification



Vérification

Après un import :

- ☐ Le dashboard est dans le bon dossier et s'ouvre sans erreur
- ☐ Les menus déroulants (job, instance...) proposent les libellés attendus
- ☐ Sur une plage d'une heure, aucun panneau n'affiche « N/A » ou « No data »
- ☐ Le croisement de métriques (étape 2) ne signale plus de manquante après les corrections éventuelles
- ☐ La page d'accueil affiche toujours la vue d'ensemble

7 Rollback

6

Supprimer un dashboard importé

Depuis le dashboard : Dashboard settings → [Delete dashboard](#).

Par l'API, à partir de l'UID visible dans l'URL :

```
curl -X DELETE -u admin:MOT_DE_PASSE \  
http://127.0.0.1:3000/api/dashboards/uid/<uid>
```

La suppression n'affecte ni la collecte ni les autres dashboards.
En cas de remplacement d'un dashboard défaillant, supprimer l'ancien après avoir validé le nouveau.

8 Dépannage

Problème	Solution
Tous les panneaux sont vides	Datasource non liée à l'import (cause n° 2 de la section 3) ou variables de gabarit vides (cause n° 3). Rouvrir Dashboard settings → Variables et contrôler la requête de chaque variable.
Quelques panneaux affichent « N/A », le reste fonctionne	Métriques renommées ou supprimées (cause n° 1). Rejouer le croisement de l'étape 2 pour les identifier, puis corriger les requêtes (étape 5).
« A dashboard with the same UID already exists »	Réimport d'un dashboard déjà présent : ajouter <code>"overwrite": true</code> au payload, ou supprimer l'ancien au préalable.
Les panneaux d'E/S disque des conteneurs restent vides	Limite connue de l'environnement : avec le pilote de stockage <i>containerd snapshotter</i> du CT 200, cAdvisor n'expose pas les métriques <code>container_fs_*</code> . Les panneaux CPU, mémoire et réseau ne sont pas affectés.
Le bouton « Import via grafana.com » échoue	Comportement attendu : le serveur n'a pas d'accès Internet. Suivre la voie hors-ligne du présent document.

9 Voir aussi

- **MO-PLT-014** : Se connecter à Grafana — accès et compte
- **MO-PLT-008** : Consulter les dashboards Grafana — lecture quotidienne des tableaux de bord
- **MO-PLT-022** : Ajouter un serveur à la supervision Prometheus — alimente les dashboards en nouvelles instances
- **MO-PLT-023** : Gérer les règles d'alerte Prometheus — la tuile « Alertes actives » de la vue d'ensemble

Références

- Grafana — *Dashboard HTTP API* (import, suppression, préférences d'organisation)
- Grafana — catalogue public de dashboards (grafana.com/grafana/dashboards)
- prometheus-community — `windows_exporter`, journal des changements (renommages de métriques des versions 0.30 et 0.31)
- Dashboards déployés : #23942 (Windows 2025, v0.31+), #21743 (cAdvisor Overview), #1860 (Node Exporter Full)