🎓 Monitorer avec Grafana (Grafana 13)

Support de cours complet

1. Module 0 : Accueil et préparation — 15 min
2. Module 1 : Introduction à la supervision — 45 min
3. Module 2 : Grafana : architecture, installation et configuration — 1 h
4. Module 3 : Connecteurs et sources de données — 1 h 30
5. Module 4 : Création d'un tableau de bord — 1 h 15
6. Module 5 : Mise en place des graphiques — 1 h 45
7. Module 6 : Tableaux de bord paramétrables — 1 h
8. Module 7 : Utilisation des plugins — 1 h
9. Module 8 : Partage du tableau de bord — 45 min
10. Module 9 : Gestion des alertes — 1 h 30
11. Module 10 : Utilisateurs, équipes et droits — 45 min
12. Module 11 : Croisement de données — 40 min
13. Module 12 : Développement de plugin et bilan — 25 min

🎓 Module 0 : Accueil et préparation

Durée
15 min (S0.1 : 15 min)

Prérequis
identifiants de la VM reçus

Fil rouge
présentation de *Telcéa*

Objectifs

- Situer les 2 jours de formation et leurs résultats attendus
- Décrire le fil rouge *Telcéa* et son état final visé
- Accéder à Grafana et au bureau de sa VM (2 URL, 1 mot de passe)
- Évaluer son niveau de départ (auto-positionnement)

Déroulé

- Séquence S0.1 — Accueil, fil rouge et accès aux VM (15 min)

Séquence S0.1

Accueil, fil rouge et accès aux VM

15 min

1. Grafana en 2 jours
2. Le programme en un schéma
3. Les 12 modules
4. Fil rouge : Telcéa
5. Fil rouge : le réseau supervisé
6. Environnement de travail
7. Méthode de travail

🕮 Grafana en 2 jours

Construire (J1), puis exploiter (J2)

```
Formation = 2 jours × 7 h = 14 h (≈ 65 % de pratique)
```

JOUR	THÈME	RÉSULTAT
J1	Construire	Sources raccordées, tableau de bord du NOC
J2	Exploiter	Variables, plugins, partage, alertes, droits

📌 Validation

QCM d'ancrage par module, QCM final en fin de J2

📖 Le programme en un schéma

Chaque module s'appuie sur le résultat du précédent

- J1 : M01 à M05 · J2 : M06 à M12
- Bilan QCM en fin de J1, QCM final en fin de J2

Les 12 modules

Durées de contenu, pauses non comprises

JOUR	MODULES	DURÉE
J1	M01 Introduction · M02 Grafana · M03 Sources de données	3 h 15
J1	M04 Création d'un dashboard · M05 Graphiques · bilan J1	3 h 00
J2	M06 Variables · M07 Plugins · M08 Partage	2 h 45
J2	M09 Alertes · M10 Droits · M11 Croisement · M12 Plugin, bilan	3 h 20

🕮 Fil rouge : Telcéa

Opérateur mobile et fibre fictif, supervisé depuis son NOC

Réseau = 12 sites radio (4G / 5G) + 4 régions + 3 liens vers le cœur

⇄ Avant / Après

Avant: indicateurs dispersés → alarmes lues à la main → panne vue par les clients
Après: tableau de bord national → alertes par équipe → panne vue en 1 à 5 min

- Deux équipes : `radio` (sites) et `transport` (liens, routeurs)
- Un incident au J2 : tempête en Bretagne (sites coupés, lien saturé)

🏢 En entreprise

NOC = *Network Operations Center* (supervision 24 h / 24)

🕮 Fil rouge : le réseau supervisé

4 régions reliées au cœur de réseau parisien

- Données réseau : simulateur Telcéa → Prometheus
- Données système de la VM : Telegraf → InfluxDB

🕮 Environnement de travail

Une VM par stagiaire, accessible depuis un simple navigateur

Accès = 1 URL Grafana + 1 URL bureau + 1 mot de passe

ACCÈS	URL	USAGE
Grafana	`https://stagiaireNN.<SESSION>.kovalibre-labs.com/grafana/`	Tous les labs
Bureau Linux	`https://stagiaireNN.<SESSION>.kovalibre-labs.com/`	Terminal (labs ⭐⭐⭐)

- Identifiant Grafana : `admin` · identifiant bureau : `stagiaire`
- Mot de passe identique pour les deux (fourni par la formatrice)

⚠ Piège courant

oubli de `/grafana/` en fin d'URL → page du bureau Linux

🕮 Méthode de travail

Théorie courte, pratique immédiate

- ⭐ Lab guidé : étapes fournies, résultat attendu indiqué
- ⭐⭐ Exercice semi-autonome : objectif + indices, sans les étapes
- ⭐⭐⭐ Défi autonome : problème ouvert (pour les plus rapides)
- 🚀 Bonus : extension facultative
- Corrections publiées après chaque lab

💡 Bonne pratique

lab bloqué > 5 min → signaler (tableau de bord de rattrapage disponible)

À retenir

Points clés

- 2 jours = Construire (J1) → Exploiter (J2)
- Fil rouge *Telcéa* : 12 sites, 4 régions, 3 liens, 2 équipes
- Une VM par stagiaire : Grafana en `/grafana/`, bureau à la racine
- Un seul mot de passe pour Grafana et le bureau

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quel est le rôle principal de Grafana ?

- **A.** Stocker les métriques
- **B.** Collecter les journaux des serveurs
- **C.** Visualiser et alerter sur des données stockées ailleurs
- **D.** Remplacer une base de données

Réponse C — Grafana interroge des sources externes pour visualiser et alerter ; il ne stocke pas les mesures.

Révéler la réponse

2. Qu'est-ce qu'une série temporelle ?

- **A.** Une liste de valeurs horodatées pour une même mesure
- **B.** Un tableau de bord planifié
- **C.** Un fichier de journaux
- **D.** Une requête SQL

Réponse A — Série temporelle = suite de couples (horodatage, valeur) pour une même mesure et les mêmes étiquettes.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Quel outil de supervision interroge périodiquement des cibles pour collecter des métriques ?

- **A.** Grafana
- **B.** Excel
- **C.** Mailpit
- **D.** Prometheus

Réponse D — Prometheus collecte en mode *pull* : lecture périodique des pages `/metrics` des cibles.

Révéler la réponse

4. Quel indicateur mesure le mieux la disponibilité d'un site radio ?

- **A.** Le nombre d'abonnés en fin de mois
- **B.** Le pourcentage de temps où le site est en service
- **C.** La température de la salle
- **D.** Le nom du site

Réponse B — Disponibilité = part du temps en service (ex. 99,9 % sur un mois).

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Que signifie NOC ?

- **A.** Network Operations Center
- **B.** New Online Console
- **C.** Node Output Collector
- **D.** Network Optical Cable

Réponse A — Network Operations Center : centre de supervision du réseau.

Révéler la réponse

6. Pourquoi définir des seuils sur un indicateur ?

- **A.** Pour changer l'unité affichée
- **B.** Pour réduire la taille de la base
- **C.** Pour trier les tableaux de bord
- **D.** Pour signaler visuellement ou par alerte un état anormal

Réponse D — Un seuil transforme une valeur en état (normal, attention, critique), affiché par une couleur ou une alerte.

Révéler la réponse

🎓 Module 1 : Introduction à la supervision

Durée
45 min (S1.1 : 25 min · S1.2 : 20 min)

Prérequis
M00

Fil rouge
NOC de Telcéa, aucun outil raccordé

Objectifs

- Expliquer le rôle de la supervision dans une architecture applicative et une démarche DevOps
- Distinguer métriques, journaux et traces, et les modes de collecte *pull* / *push*
- Situer Grafana, Prometheus et InfluxDB parmi les outils de supervision
- Lister les indicateurs clés à superviser pour un opérateur réseau

Déroulé

- Séquence S1.1 — Architectures, DevOps et supervision (25 min)
- Séquence S1.2 — Tour d'horizon des outils (20 min)

Séquence S1.1

Architectures, DevOps et supervision

25 min

1. Des architectures plus difficiles à surveiller
2. L'univers DevOps
3. La supervision : définition
4. Pourquoi superviser ?
5. Que superviser chez un opérateur ?
6. Trois types de données
7. Collecter : *pull* ou *push* ?

Des architectures plus difficiles à surveiller

Plus de composants = plus de points de panne

Évolution = Monolithe → N-tiers → Microservices + conteneurs

- Monolithe : un processus à surveiller (simple, mais tout tombe d'un coup)
- Microservices : panne partielle possible (un service lent dégrade les autres)
- Conteneurs : durée de vie de quelques minutes (adresses et noms changent)

🕮 L'univers DevOps

Développement + Exploitation : un objectif commun, le service rendu

DevOps = Culture + Automatisation + Mesure + Partage

- Supervision = fin ET début de la boucle (les mesures guident la suite)
- Mesure partagée : mêmes tableaux de bord pour les développeurs et l'exploitation

🕮 La supervision : définition

S'assurer que tout fonctionne comme attendu, et le prouver

```
Supervision = Collecter + Stocker + Visualiser + Alerter
```

- Collecter : relever les mesures des équipements et applications
- Stocker : historiser (tendances, comparaison jour / semaine)
- Visualiser : tableaux de bord partagés, en temps réel
- Alerter : prévenir la bonne équipe avant l'utilisateur

⇄ Avant / Après

Avant: panne signalée par un client → recherche manuelle (heures)
Après: alerte automatique → tableau de bord de diagnostic (minutes)

Pourquoi superviser ?

Trois usages, trois publics

USAGE	EXEMPLE TELCÉA	PUBLIC
Comprendre et planifier	Croissance du trafic 5G par région	Pilotage, architecture
Visualiser et partager	Écran du NOC : état des 12 sites	Exploitation
Alerter et corriger	Site BRE-01 hors service → équipe radio	Astreinte

En entreprise

capacité planifiée sur l'historique (ex. ajout d'une cellule avant saturation)

❓ Que superviser chez un opérateur ?

Disponibilité, capacité, qualité de service

`Indicateur = Mesure + Seuil + Responsable`

FAMILLE	INDICATEUR TELCÉA	UNITÉ
Disponibilité	Site en service (`telcea_site_up`)	0 / 1, % du temps
Capacité	Trafic par cellule, utilisation des liens	octets/s, %
Qualité	Latence, perte de paquets, réussite d'appel	ms, %, %
Environnement	Température, bascule sur batterie	°C, 0 / 1

🕮 Trois types de données

Chiffres, textes, parcours

`Observabilité = Métriques + Journaux + Traces`

TYPE	FORME	EXEMPLE TELCÉA	OUTIL TYPE
Métrique	Valeur numérique horodatée	914 abonnés sur PAR-01	Prometheus, InfluxDB
Journal (log)	Ligne de texte horodatée	`Bascule sur batterie` (REN-01)	Loki, Elasticsearch
Trace	Parcours d'une requête	Appel API → base → réponse	Tempo, Jaeger

- Série temporelle = suite de couples (horodatage, valeur) pour une même mesure
- Formation centrée sur les métriques (et les journaux d'alarmes au M09)

? Collecter : *pull* ou *push* ?

Qui prend l'initiative de l'envoi

- *Pull* : la base interroge les cibles (cible muette = détectée tout de suite)
- *Push* : l'agent envoie ses mesures (adapté aux équipements derrière un pare-feu)

Séquence S1.2

Tour d'horizon des outils

20 min

1. La chaîne de supervision
2. Outils historiques et outils modernes
3. Grafana : le point d'entrée unique
4. Prometheus et InfluxDB en bref

🕮 La chaîne de supervision

Trois rôles, souvent trois outils

Chaîne = Collecteur → Base de séries temporelles → Visualisation

- Grafana ne stocke AUCUNE mesure : il interroge les bases à chaque affichage

🕮 Outils historiques et outils modernes

Du contrôle d'état à la série temporelle

OUTIL	MODÈLE	POINT FORT
Nagios, Centreon	Contrôles OK / WARNING / CRITICAL	Supervision d'états, parc réseau
Zabbix	Agents + SNMP, base SQL	Tout-en-un (collecte, alertes)
Prometheus	*Pull*, séries temporelles, PromQL	Conteneurs, Kubernetes, alerting
InfluxDB	*Push*, séries temporelles, SQL	Mesures à haute fréquence, IoT
Elastic Stack	Indexation de documents	Recherche dans les journaux
Datadog	SaaS tout-en-un, facturé à l'usage	Aucune infrastructure à gérer

🕮 Grafana : le point d'entrée unique

Visualiser, explorer, alerter, quelles que soient les sources

```
Grafana = Sources de données + Tableaux de bord + Alertes
```

- Logiciel libre (licence AGPLv3 depuis la version 8, éditeur Grafana Labs)
- Trois éditions : OSS (formation), Enterprise (rapports PDF, SSO avancé), Cloud (SaaS)
- Sources de données par plugins (Prometheus, InfluxDB, SQL, Zabbix, Elasticsearch)
- Version de la formation : **13.3.1** (novembre 2026)

⚠ Piège courant

« Grafana ne remonte rien » → le problème est presque toujours dans la source

📖 Prometheus et InfluxDB en bref

Les deux bases de la formation

	PROMETHEUS 3.16	INFLUXDB 3.12 CORE
Collecte	*Pull* (lecture des cibles)	*Push* (Telegraf, API)
Langage	PromQL	SQL (et InfluxQL)
Alertes	Règles + Alertmanager	Non (passer par Grafana)
Données Telcéa	Réseau simulé + VM + conteneurs	Mesures système de la VM

💡 Bonne pratique

un outil par usage, Grafana pour tout afficher au même endroit

À retenir

Points clés

- `Supervision = Collecter + Stocker + Visualiser + Alerter`
- Métriques (chiffres), journaux (textes), traces (parcours)
- *Pull* : Prometheus lit les cibles · *Push* : Telegraf écrit dans InfluxDB
- Grafana interroge, ne stocke pas : une panne d'affichage vient souvent de la source

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quelle étape de la boucle DevOps alimente directement la planification suivante ?

- **A.** Construire
- **B.** Superviser
- **C.** Tester
- **D.** Coder

Réponse B — Superviser ferme la boucle DevOps : les mesures orientent la planification suivante.

Révéler la réponse

2. « Bascule sur batterie du site REN-01 » est un exemple de :

- **A.** Journal
- **B.** Métrique
- **C.** Trace
- **D.** Seuil

Réponse A — Ligne de texte horodatée = journal (log).

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Quel mode de collecte utilise Prometheus ?

- **A.** *Push* : les cibles envoient leurs mesures
- **B.** Import manuel de fichiers CSV
- **C.** Requêtes SNMP uniquement
- **D.** *Pull* : Prometheus lit périodiquement les cibles

Réponse D — Prometheus lit périodiquement l'adresse `/metrics` des cibles (*pull*).

Révéler la réponse

4. Où sont stockées les mesures affichées dans un tableau de bord Grafana ?

- **A.** Dans la base interne de Grafana
- **B.** Dans le navigateur
- **C.** Dans les sources de données (Prometheus, InfluxDB)
- **D.** Dans les fichiers de journaux

Réponse C — Grafana interroge les sources à chaque affichage ; sa base interne ne contient que la configuration.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Quel outil est historiquement fondé sur des contrôles d'état OK / WARNING / CRITICAL ?

- **A.** Prometheus
- **B.** Nagios
- **C.** InfluxDB
- **D.** Grafana

Réponse B — Nagios : contrôles actifs aux états OK / WARNING / CRITICAL / UNKNOWN.

Révéler la réponse

6. Pour Telcéa, « latence et perte de paquets » relèvent de quelle famille d'indicateurs ?

- **A.** Disponibilité
- **B.** Capacité
- **C.** Environnement
- **D.** Qualité de service

Réponse D — Latence et perte de paquets mesurent la qualité du service rendu.

Révéler la réponse

🎓 Module 2 : Grafana : architecture, installation et configuration

Durée
1 h (S2.1 : 25 min · S2.2 : 35 min)

Prérequis
M01

Fil rouge
Grafana 13.3.1 démarré sur la VM, aucune source raccordée

Objectifs

- Décrire l'architecture de Grafana et le parcours d'une requête jusqu'au panel
- Choisir un mode d'installation (Docker, paquet, Kubernetes) selon le contexte
- Modifier la configuration par fichier et par variable `GF_*`, et vérifier la valeur effective
- Citer les principales nouveautés de Grafana 12 et 13

Déroulé

- Séquence S2.1 — Architecture, fonctionnement, installation (25 min)
- Séquence S2.2 — Configuration, administration, nouveautés (35 min)

Séquence S2.1

Architecture, fonctionnement, installation

25 min

1. Comment Grafana est construit
2. Le parcours d'une requête
3. Installer Grafana
4. Installer avec Docker
5. La stack Telcéa sur chaque VM

🕮 Comment Grafana est construit

Un serveur, une base de configuration, des plugins

```
Grafana = Serveur (Go) + Interface (React) + Base interne + Plugins
```


- Base interne : utilisateurs, tableaux de bord, alertes (JAMAIS les mesures)
- SQLite par défaut ; MySQL ou PostgreSQL pour plusieurs serveurs Grafana

🕮 Le parcours d'une requête

Du clic à la courbe, à chaque rafraîchissement

Panel = Requête + Plage de temps → Data frame → Visualisation

- Data frame = format commun (colonnes temps + valeurs), quelle que soit la source
- Une requête par panel et par rafraîchissement (5 s × 20 écrans = 240 requêtes / min)

Installer Grafana

Quatre modes, un même logiciel

MODE	COMMANDE OU SOURCE	USAGE
Conteneur Docker	`grafana/grafana:13.3.1`	Tests, petites équipes (formation)
Paquet Linux	Dépôt APT / RPM `apt.grafana.com`	Serveur dédié, service `grafana-server`
Kubernetes	Chart Helm ou Grafana Operator 5.26	Plateformes conteneurisées
Grafana Cloud	Offre SaaS de Grafana Labs	Aucune infrastructure à gérer

- Première connexion : `admin` / `admin`, changement de mot de passe imposé
- Formation : conteneur Docker, mot de passe défini par `GF_SECURITY_ADMIN_PASSWORD`

Installer avec Docker

Une commande, un volume pour conserver la base interne

```bash
docker run -d --name grafana -p 3000:3000 \
  -v grafana_data:/var/lib/grafana \
  -e GF_SECURITY_ADMIN_PASSWORD=<MOT_DE_PASSE> \
  grafana/grafana:13.3.1
```

Contrôle :

```bash
curl -s http://localhost:3000/api/health
```

```
{
  "database": "ok",
  "version": "13.3.1",
  "commit": "da3aa3ff574e7dbbf808c66b6daa744defed0441"
}
```

- Version figée (`13.3.1`), jamais `latest` en production

🕮 La stack Telcéa sur chaque VM

Grafana + bases + collecteurs, décrits dans `compose.yaml`

- Collecteurs : simulateur Telcéa, node-exporter, cAdvisor, Telegraf, Alloy
- Courriels : Mailpit (`/mail/`), aucun courriel ne sort de la VM

Séquence S2.2

Configuration, administration, nouveautés

35 min

1. Où se règle Grafana ?
2. Vérifier la configuration effective
3. Le provisioning
4. Un fournisseur de tableaux de bord
5. Administrer Grafana
6. Nouveautés de Grafana 12 et 13

❓ Où se règle Grafana ?

Trois niveaux, le dernier l'emporte

`Valeur effective = defaults.ini < grafana.ini < variables GF_*`

- `defaults.ini` : valeurs par défaut (ne JAMAIS modifier)
- `grafana.ini` : fichier de l'installation (`/etc/grafana/grafana.ini` en paquet)
- Variable `GF_<SECTION>_<CLÉ>` : surcharge sans toucher aux fichiers (Docker)

SECTION	CLÉ	VARIABLE
`[server]`	`root_url`	`GF_SERVER_ROOT_URL`
`[security]`	`admin_password`	`GF_SECURITY_ADMIN_PASSWORD`
`[smtp]`	`host`	`GF_SMTP_HOST`

📖 Vérifier la configuration effective

Administration → General → Settings

Settings = toutes les sections + valeurs réellement appliquées

SECTION	CLÉ	VALEUR SUR LA VM
`server`	`root_url`	`https://stagiaireNN.<SESSION>…/grafana/`
`database`	`type`	`sqlite3`
`smtp`	`enabled`	`true` (vers Mailpit)
`paths`	`provisioning`	`/etc/grafana/provisioning`

💡 Bonne pratique

toujours vérifier ici après un changement (faute de frappe = valeur ignorée)

🕮 Le provisioning

Configurer Grafana par fichiers, au démarrage

Provisioning = Fichiers YAML + Redémarrage → Ressources créées

- Dossiers : `datasources`, `dashboards`, `plugins`, `alerting`
- Ressource provisionnée = non modifiable dans l'interface (source de vérité : le fichier)

🕮 Un fournisseur de tableaux de bord

Fichier `provisioning/dashboards/noc.yaml`

```yaml
apiVersion: 1
providers:
  - name: noc
    folder: NOC
    type: file
    allowUiUpdates: false
    options:
      path: /etc/grafana/provisioning/dashboards/noc
```

- Chaque fichier JSON du dossier `noc/` = un tableau de bord dans le dossier « NOC »
- Fichier supprimé = tableau de bord supprimé au redémarrage suivant

🕮 Administrer Grafana

Menu Administration : tout ce qui concerne l'instance

RUBRIQUE	CONTENU
General → Settings, Stats	Configuration effective, compteurs
Users and access	Utilisateurs, équipes, comptes de service (M10)
Plugins and data → Plugins	Plugins installés et catalogue (M07)
Advisor	Contrôles de santé (sources, plugins obsolètes)

- Préférences de l'organisation : nom, thème, fuseau horaire, page d'accueil
- Préférences du profil : langue de l'interface (dont Français), thème

🕮 Nouveautés de Grafana 12 et 13

Ce qui change par rapport aux versions 8 et 9

VERSION	NOUVEAUTÉ	MODULE
12.0 (2025)	Applications Drilldown	M03
12.0	Plugins Angular supprimés	M07
13.0 (2026)	Tableaux de bord dynamiques	M04
13.0	Restauration après suppression	M06
13.0	Advisor, jauge, suggestions	M02, M05
13.0	`grafana-cli`, Image Renderer retirés	M07, M08

À retenir

Points clés

- Grafana = serveur + interface + base de configuration + plugins (aucune mesure stockée)
- `Valeur effective = defaults.ini < grafana.ini < GF_*`, vérifiée dans Settings
- Provisioning : fichiers YAML lus au démarrage, ressources verrouillées dans l'interface
- Grafana 13 : tableaux de bord dynamiques, `grafana cli` (sans tiret), plus d'Image Renderer

QCM d'ancrage (1/3)

Questions 1 à 2

1. Que contient la base interne de Grafana ?

A. Les mesures de Prometheus

B. Les journaux des conteneurs

C. Les utilisateurs, tableaux de bord et alertes

D. Les images des panels

Réponse C — La base interne contient la configuration (utilisateurs, tableaux de bord, alertes), jamais les mesures.

Révéler la réponse

2. Quelle variable surcharge la clé `root_url` de la section `[server]` ?

A. `GRAFANA_ROOT_URL`

B. `GF_SERVER_ROOT_URL`

C. `GF_ROOT_URL`

D. `SERVER_ROOT_URL`

Réponse B — `GF_<SECTION>_<CLÉ>` en majuscules : `GF_SERVER_ROOT_URL`.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Un fichier `grafana.ini` et une variable `GF_*` définissent la même clé : laquelle s'applique ?

- **A.** Le fichier `grafana.ini`
- **B.** `defaults.ini`
- **C.** La première lue au démarrage
- **D.** La variable `GF_*`

Réponse D — Les variables d'environnement l'emportent sur les fichiers `.ini`.

Révéler la réponse

4. Où vérifier la valeur réellement appliquée d'un paramètre ?

- **A.** Administration → General → Settings
- **B.** Profil → Préférences
- **C.** Explore
- **D.** Dans `defaults.ini`

Réponse A — **Settings** affiche la configuration effective, fusion de toutes les sources.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Que se passe-t-il pour un tableau de bord provisionné modifié dans l'interface ?

- **A.** Il est enregistré dans le fichier JSON
- **B.** Grafana redémarre
- **C.** L'enregistrement est refusé (le fichier reste la référence)
- **D.** Il est dupliqué automatiquement

Réponse C — Ressource provisionnée : le fichier fait foi, l'enregistrement depuis l'interface est refusé.

Révéler la réponse

6. Quelle commande installe un plugin sur Grafana 13 ?

- **A.** `grafana-cli plugins install <ID>`
- **B.** `grafana cli plugins install <ID>`
- **C.** `grafana-server install <ID>`
- **D.** `apt install grafana-plugin-<ID>`

Réponse B — Grafana 13 a supprimé `grafana-cli` ; commande : `grafana cli plugins install <ID>`.

Révéler la réponse

🎓 Module 3 : Connecteurs et sources de données

Durée
1 h 30 (S3.1 : 45 min · S3.2 : 45 min)

Prérequis
M02

Fil rouge
Grafana configuré, aucune source raccordée

Objectifs

- Raccorder une source Prometheus et une source InfluxDB 3 (SQL), testées avec succès
- Écrire des requêtes PromQL de base : sélection, filtre, `rate()`, agrégation par étiquette
- Écrire une requête SQL InfluxDB 3 filtrée sur la plage de temps de Grafana
- Retrouver dans Explore une mesure donnée du réseau Telcéa et de la VM

Déroulé

- Séquence S3.1 — Sources de données et initiation PromQL (45 min)
- Séquence S3.2 — InfluxDB 3 et exploration des données (45 min)

Séquence S3.1

Sources de données et initiation PromQL

45 min

1. Une source de données ?
2. Familles de sources
3. Raccorder Prometheus
4. Le modèle de données Prometheus
5. Trois types de métriques
6. PromQL : sélectionner et filtrer
7. PromQL : débit et agrégation

? Une source de données ?

La configuration qui permet à Grafana d'interroger une base

Source de données = Plugin + URL + Authentification

- Une source = une base ; plusieurs sources par instance (sans limite)
- Plugins fournis : Prometheus, InfluxDB, Loki, MySQL, PostgreSQL, Elasticsearch

Familles de sources

Le plugin dicte le langage de requête

FAMILLE	EXEMPLES	LANGAGE
Séries temporelles	Prometheus, InfluxDB, Graphite	PromQL, SQL / InfluxQL
Journaux	Loki, Elasticsearch	LogQL, Lucene
Bases SQL	MySQL, PostgreSQL, SQL Server	SQL
Supervision existante	Zabbix (plugin communautaire)	Éditeur visuel
Cloud	CloudWatch, Azure Monitor	Éditeurs dédiés

En entreprise

Zabbix ou Centreon conservés, Grafana ajouté pour les tableaux de bord

🕮 Raccorder Prometheus

Connections → Data sources → Add new data source → Prometheus

CHAMP	VALEUR TELCÉA
Name	`Prometheus`
Prometheus server URL	`http://prometheus:9090`
Default	Activé

Résultat de **Save & test** :

```
Successfully queried the Prometheus API.
```

⚠ Piège courant

`localhost` = le conteneur Grafana lui-même → nom du service (`prometheus`)

Le modèle de données Prometheus

Une série = un nom + des étiquettes

Série = Nom de métrique + {Étiquettes} → (horodatage, valeur)

```
telcea_cell_users{region="Ile-de-France", site="PAR-01", techno="5G"} 914
```

- Nom : ce qui est mesuré (`telcea_cell_users` : abonnés connectés)
- Étiquettes : dimensions pour filtrer et regrouper (`region` , `site` , `techno`)
- Ajouts automatiques : `job` (nom de la collecte), `instance` (cible lue)
- Une combinaison d'étiquettes = une série distincte (12 sites = 12 séries)

🕮 Trois types de métriques

Le type dicte la fonction à utiliser

TYPE	COMPORTEMENT	EXEMPLE TELCÉA	REQUÊTE TYPE
Gauge	Monte et descend	`telcea_cell_users`	Valeur brute
Counter	Ne fait que croître	`telcea_cell_traffic_bytes_total`	`rate(...[5m])`
Histogram	Répartition en tranches	`telcea_site_latency_seconds_bucket`	`histogram_quantile()`

- Counter : suffixe `_total` par convention ; seul le débit (`rate`) a du sens
- Histogram : abordé en défi (latence au 95e centile)

🕮 PromQL : sélectionner et filtrer

Le nom seul, puis des filtres sur les étiquettes

```promql
telcea_site_up                              # 12 séries : un site = une série
telcea_site_up{region="Bretagne"}           # égalité : 3 sites
telcea_site_up{region!="Bretagne"}          # différence : 9 sites
telcea_cell_users{site=~"PAR-.*|SDN-01"}    # regex : Paris + Saint-Denis
telcea_site_up == 0                         # comparaison : sites hors service
```

- Opérateurs : `=`, `!=`, `=~` (regex), `!~` (regex exclue)
- Valeurs `telcea_site_up` : `1` = en service, `0` = hors service

🕮 PromQL : débit et agrégation

Transformer un compteur en débit, regrouper par région

Débit = rate(compteur[fenêtre]) · Total = sum by (étiquette) (...)

```promql
rate(telcea_cell_traffic_bytes_total{direction="down"}[5m]) * 8
sum by (region) (telcea_cell_users)
sum by (region) (rate(telcea_cell_traffic_bytes_total{direction="down"}[5m])) * 8
```

Sortie (banc, 18/11/2026 10 h, abonnés par région) :

```
{region="Bretagne"} 989        {region="Ile-de-France"} 2529
{region="Hauts-de-France"} 1032   {region="Provence"} 1386
```

💡 Bonne pratique

`rate()` toujours AVANT `sum` (sinon redémarrage de cible = pic faux)

Séquence S3.2

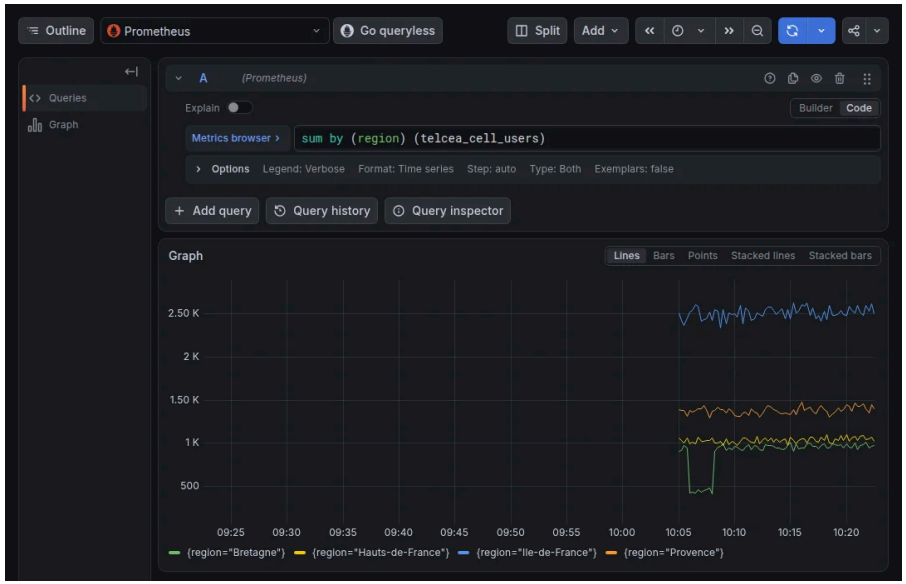
InfluxDB 3 et exploration des données

45 min

1. Explore : interroger sans tableau de bord
2. Explore en pratique
3. InfluxDB 3 : le modèle de données
4. Raccorder InfluxDB 3 en SQL
5. Raccorder InfluxDB 3 : les valeurs
6. SQL InfluxDB 3 dans Grafana
7. PromQL ou SQL : la même question

🕮 Explore : interroger sans tableau de bord

Menu Explore : une source, une requête, un graphe



Explore en pratique

Construire, comparer, retrouver ses requêtes

FONCTION	USAGE
Builder / Code	Requête guidée ou saisie directe (PromQL)
Metrics browser	Liste des métriques et de leurs étiquettes
Query history	Requêtes déjà exécutées (réutilisables)
Split	Deux requêtes côte à côte (deux sources possibles)
Add → Add to dashboard	Requête envoyée vers un tableau de bord (M04)

- Plage de temps (en haut à droite) : s'applique à TOUTES les requêtes

InfluxDB 3 : le modèle de données

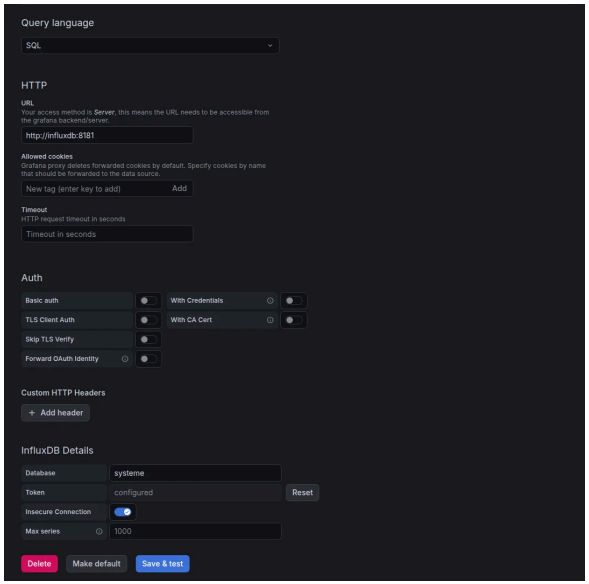
Bases, tables, colonnes : du SQL pour les séries temporelles

`InfluxDB 3 = Base + Tables (mesures) + Colonnes (tags, champs, time)`

NOTION	TELCÉA (BASE `SYSTEME`, ÉCRITE PAR TELEGRAF)
Base (*database*)	`systeme`
Table (*measurement*)	`cpu`, `mem`, `disk`, `net`, `system`
Tag (texte indexé)	`host` = `vm-noc`, `cpu` = `cpu-total`
Champ (valeur)	`usage_user`, `usage_idle`, `available_percent`
Temps	Colonne `time` (UTC)

Raccorder InfluxDB 3 en SQL

Connections → Data sources → Add new data source → InfluxDB



🕮 Raccorder InfluxDB 3 : les valeurs

Cinq champs, dans cet ordre

CHAMP	VALEUR TELCÉA
Query language	`SQL` (à choisir EN PREMIER)
URL	`http://influxdb:8181`
Database	`systeme`
Token	`INFLUX_TOKEN` de `stack/.env`
Insecure Connection	Activé (pas de TLS dans la stack)

Résultat de **Save & test** : `OK`

⚠ Piège courant

Insecure Connection désactivé → échec de connexion (TLS attendu)

🕮 SQL InfluxDB 3 dans Grafana

Des macros pour suivre la plage de temps et l'intervalle

```sql
SELECT $__dateBin(time) AS time, avg(usage_user) AS cpu_utilisateur
FROM cpu
WHERE $__timeFilter(time)
GROUP BY 1
ORDER BY 1
```

- `$__timeFilter(time)` : filtre sur la plage de temps choisie dans Grafana
- `$__dateBin(time)` : regroupement par intervalle (alias `AS time` OBLIGATOIRE)
- Format **Time series** : une courbe par colonne de valeur

⚠ Piège courant

sans `AS time` → erreur `no time column found`

🕮 PromQL ou SQL : la même question

Utilisation CPU de la VM, deux sources

SOURCE	REQUÊTE
Prometheus (node-exporter)	`100 * (1 - avg(rate(node_cpu_seconds_total{mode="idle"}[5m])))`
InfluxDB (Telegraf)	`SELECT $__dateBin(time) AS time, avg(100 - usage_idle) FROM cpu ...`

- Deux collecteurs, deux bases, des valeurs proches (pas identiques : fenêtres différentes)
- Choix de la source : selon l'équipe propriétaire et l'historique disponible

À retenir

Points clés

- `Source de données = Plugin + URL + Authentification` ; URL vue PAR LE SERVEUR Grafana
- `Série = Nom + {Étiquettes}` ; gauge → valeur, counter → `rate()`, puis `sum by`
- InfluxDB 3 : SQL + macros `$__timeFilter` et `$__dateBin(time) AS time`
- Explore = laboratoire de requêtes avant le tableau de bord

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quelle URL saisir pour la source Prometheus dans la stack Telcéa ?

- **A.** `http://prometheus:9090`
- **B.** `http://localhost:9090`
- **C.** `https://prometheus.telcea.example`
- **D.** `http://grafana:3000`

Réponse A — URL résolue par le serveur Grafana dans le réseau Docker : nom du service.

Révéler la réponse

2. `telcea_cell_traffic_bytes_total` est un compteur : quelle requête donne un débit ?

- **A.** `telcea_cell_traffic_bytes_total`
- **B.** `sum(telcea_cell_traffic_bytes_total)`
- **C.** `rate(telcea_cell_traffic_bytes_total[5m])`
- **D.** `avg(telcea_cell_traffic_bytes_total)`

Réponse C — Un compteur ne fait que croître : `rate()` donne le débit par seconde.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Quel filtre sélectionne les sites dont le nom commence par `PAR-` ?

- **A.** `{site="PAR-*"}`
- **B.** `{site=~"PAR-.*"}`
- **C.** `{site!="PAR-"}`
- **D.** `{site~"PAR"}`

Réponse B — `=~` applique une expression régulière : `PAR-.*`.

Révéler la réponse

4. Que fait `sum by (region) (telcea_cell_users)` ?

- **A.** Additionne les abonnés de tous les sites en une seule valeur
- **B.** Calcule la moyenne par région
- **C.** Filtre la région `region`
- **D.** Additionne les abonnés par région (une série par région)

Réponse D — `sum by (region)` additionne en conservant une série par valeur de `region`.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Quelle macro limite une requête SQL InfluxDB à la plage de temps de Grafana ?

- **A.** `$__interval`
- **B.** `$__range`
- **C.** `$__timeFilter(time)`
- **D.** `now() - 1h`

Réponse C — `$__timeFilter(time)` traduit la plage de temps du tableau de bord en clause `WHERE`.

Révéler la réponse

6. Quelle option activer pour la source InfluxDB de la stack Telcéa ?

- **A.** Insecure Connection
- **B.** Basic auth
- **C.** Skip TLS Verify
- **D.** With Credentials

Réponse A — Flight SQL sans TLS dans la stack : **Insecure Connection** obligatoire.

Révéler la réponse

🎓 Module 4 : Création d'un tableau de bord

Durée
1 h 15 (S4.1 : 35 min · S4.2 : 40 min)

Prérequis
M03 (sources Prometheus et InfluxDB)

Fil rouge
requêtes validées dans Explore, aucun tableau de bord

Objectifs

- Appliquer une convention de nommage et ranger un tableau de bord dans un dossier
- Créer un tableau de bord organisé en lignes, avec ses réglages (description, tags, rafraîchissement)
- Choisir le type de widget adapté à une question de supervision
- Construire le tableau de bord « [TLC] Réseau — Vue nationale » du NOC

Déroulé

- Séquence S4.1 — Nommage, paramétrage, organisation (35 min)
- Séquence S4.2 — Types de widgets (40 min)

Séquence S4.1

Nommage, paramétrage, organisation

35 min

1. Un tableau de bord ?
2. Règles de nommage
3. Bonnes pratiques d'organisation
4. Créer un tableau de bord
5. L'éditeur de panel
6. Réglages du tableau de bord
7. Lignes et onglets
8. Plage de temps et rafraîchissement

❓ Un tableau de bord ?

Un écran de supervision, partagé et mis à jour en continu

Tableau de bord = Panels + Lignes / Onglets + Plage de temps + Variables

- Panel = une requête (ou plusieurs) + une visualisation
- Plage de temps et rafraîchissement : communs à tous les panels

📖 Règles de nommage

Retrouver un tableau de bord en 5 secondes

`Titre = [Préfixe] Domaine — Vue (+ tags + dossier)`

ÉLÉMENT	RÈGLE TELCÉA	EXEMPLE
Préfixe	Entité ou équipe propriétaire	`[TLC]`
Domaine	Périmètre supervisé	`Réseau`, `Système`
Vue	Niveau de détail	`Vue nationale`, `Détail site`
Tags	Minuscules, sans accent	`telcea`, `reseau`, `noc`
Dossier	Une équipe ou un service	`NOC Telcéa`

- Contre-exemple : « Nouveau tableau de bord (copie 3) » (introuvable dans 6 mois)

Bonnes pratiques d'organisation

Reprises du support historique, toujours valables

- Mots-clés `TEST` ou `DEV` dans le titre : jamais utilisé en exploitation
- Description renseignée (réglages) + panel texte de consignes
- Rafraîchissement adapté (30 s pour un NOC, pas 5 s)
- Variables plutôt que copies (un tableau de bord par vue, pas par site : M06)
- Revue périodique : suppression des tableaux de bord abandonnés

En entreprise

propriétaire indiqué dans la description (équipe + contact)

🕮 Créer un tableau de bord

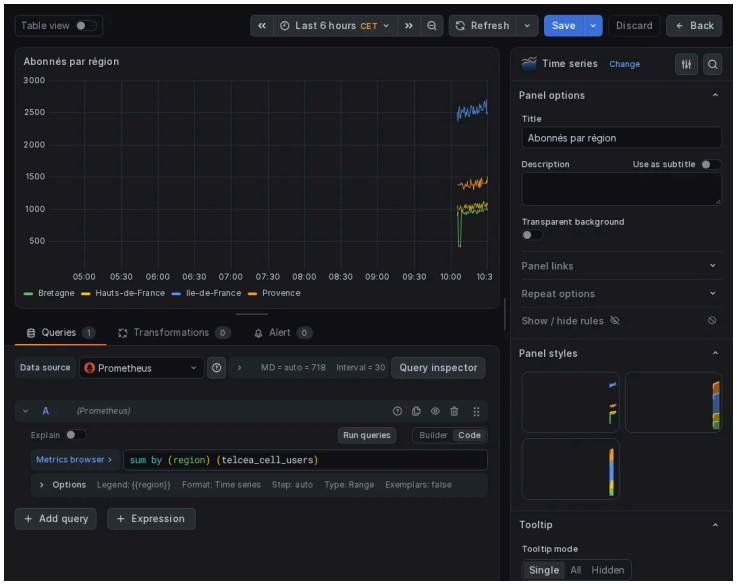
Dashboards → New → New dashboard

ÉTAPE	ACTION
1. Mise en page	**Auto grid** (tailles automatiques) ou **Custom grid** (placement libre)
2. Premier panel	Panneau **Add** → **Panel** → **Configure visualization**
3. Requête	Source, requête (copiée d'Explore), légende
4. Enregistrer	**Save** : titre, dossier, description
5. Quitter l'édition	**Exit edit** (puis **Edit** pour y revenir)

- Panneau **Add** : panel, **Add row**, **Add tab**, variable, lien

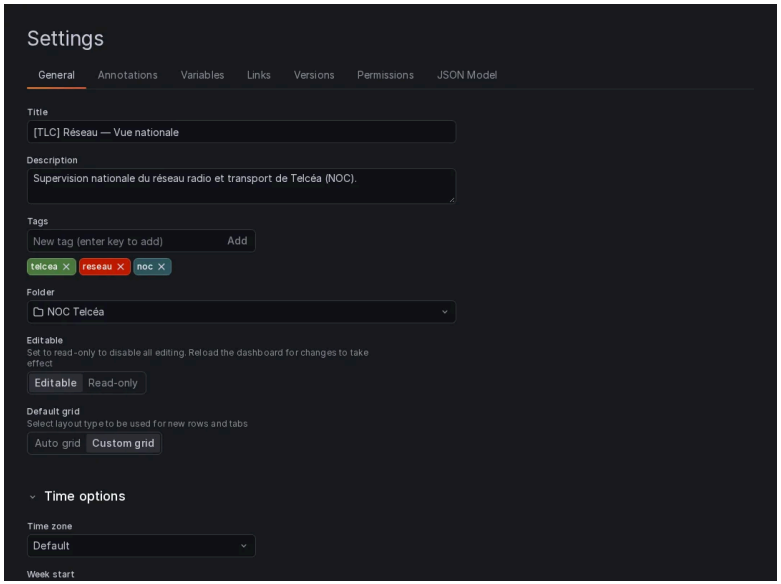
L'éditeur de panel

Requête en bas, aperçu en haut, options à droite



🕮 Réglages du tableau de bord

Icône d'engrenage (mode édition) → Settings



🕮 Lignes et onglets

Regrouper les panels par thème

ÉLÉMENT	COMPORTEMENT	USAGE TELCÉA
Ligne (*row*)	Repliable, empilée verticalement	Synthèse, Radio, Transport
Onglet (*tab*)	Un seul visible à la fois	Vue nationale / Par région
Règle d'affichage	Selon variable, plage ou résultat	Panel masqué si aucune donnée

- Ligne repliée = panels non interrogés (gain de charge sur les sources)
- Onglets et règles d'affichage : nouveautés des tableaux de bord dynamiques (Grafana 13)

🕮 Plage de temps et rafraîchissement

Barre en haut à droite, commune à tout le tableau de bord

RÉGLAGE	EXEMPLE	EFFET
Plage relative	`Last 6 hours` (`now-6h` → `now`)	Fenêtre glissante
Plage absolue	`2026-11-18 08:00` → `12:00`	Analyse d'un incident passé
Rafraîchissement	`30s`	Requêtes relancées toutes les 30 s
Fuseau horaire	Défaut de l'organisation (`Europe/Paris`)	Heure du NOC

💡 Bonne pratique

plage et rafraîchissement ENREGISTRÉS avec le tableau de bord (**Save**)

Séquence S4.2

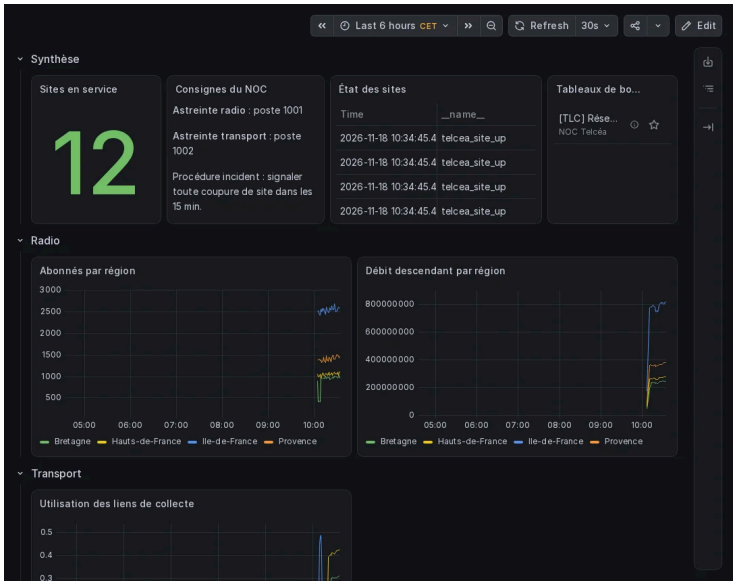
Types de widgets

40 min

1. Le tableau de bord du NOC
2. Les familles de visualisations (1/2)
3. Les familles de visualisations (2/2)
4. Choisir le bon widget

🕮 Le tableau de bord du NOC

État visé en fin de module : 3 lignes, 6 panels



🕮 Les familles de visualisations (1/2)

Courbes et chiffres

FAMILLE	VISUALISATIONS	QUESTION TYPE
Évolution dans le temps	Time series, Bar chart	« Comment évolue le trafic ? »
États dans le temps	State timeline, Status history	« Quand le site est-il tombé ? »
Distribution	Histogram, Heatmap	« Comment se répartit la latence ? »
Valeur unique	Stat, Gauge, Bar gauge	« Combien de sites en service ? »
Proportion	Pie chart	« Part de trafic par région ? »

🕮 Les familles de visualisations (2/2)

Tableaux, textes, cartes, widgets

FAMILLE	VISUALISATIONS	QUESTION TYPE
Liste détaillée	Table	« État de chaque site ? »
Texte	Text, Logs	« Que faire ? Quelles alarmes ? »
Géographie	Geomap	« Où sont les sites en panne ? »
Schéma	Canvas, Node graph	« Quel lien est saturé ? »
Widgets	Dashboard list, Alert list, News	« Quels écrans et alertes ? »

- Bouton **Change** de l'éditeur : liste complète + suggestions selon les données

🕮 Choisir le bon widget

Une question = un panel = une visualisation

- Stat pour un chiffre clé, Time series pour une tendance, Table pour le détail

À retenir

Points clés

- `Titre = [Préfixe] Domaine — Vue`, tags en minuscules, un dossier par équipe
- Dashboards → New → **Auto grid** ou **Custom grid** → **Add** → **Save** → **Exit edit**
- Lignes (repliables) et onglets (Grafana 13) pour structurer
- Une question = un panel : Stat (chiffre), Time series (tendance), Table (détail)

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quel titre respecte la convention Telcéa ?

- **A.** `Nouveau tableau de bord`
- **B.** `dashboard_reseau_v2_final`
- **C.** `TEST`
- **D.** `[TLC] Réseau — Vue nationale`

Réponse D — Préfixe d'entité, domaine, tiret cadratin, vue.

Révéler la réponse

2. Où renseigner la description et les tags d'un tableau de bord ?

- **A.** Dans le titre d'un panel
- **B.** Dans Explore
- **C.** Dans Settings, onglet General
- **D.** Dans Administration → Settings

Réponse C — Settings → General : titre, description, tags, dossier.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Quel avantage apporte une ligne repliée ?

- **A.** Les panels repliés ne sont pas interrogés
- **B.** Les panels sont supprimés
- **C.** Le tableau de bord devient public
- **D.** Les requêtes sont mises en cache 24 h

Réponse A — Panels d'une ligne repliée non interrogés : charge réduite sur les sources.

Révéler la réponse

4. Quelle visualisation pour « nombre de sites en service maintenant » ?

- **A.** Time series
- **B.** Stat
- **C.** Heatmap
- **D.** Logs

Réponse B — Stat : une valeur clé, lisible de loin.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Quelle visualisation pour « à quelle heure le site BRE-01 est-il tombé » ?

- **A.** State timeline
- **B.** Pie chart
- **C.** Gauge
- **D.** Text

Réponse A — State timeline : changements d'état dans le temps.

Révéler la réponse

6. Quelle mise en page laisse placer et dimensionner librement les panels ?

- **A.** Auto grid
- **B.** Tabs
- **C.** Playlist
- **D.** Custom grid

Réponse D — Custom grid : position et taille libres ; Auto grid : tailles uniformes automatiques.

Révéler la réponse

🎓 Module 5 : Mise en place des graphiques

Durée
1 h 45 (S5.1 : 40 min · S5.2 : 50 min · S5.3 : 15 min)

Prérequis
M04

Fil rouge
« [TLC] Réseau — Vue nationale » (3 lignes, valeurs brutes)

Objectifs

- Choisir et configurer les visualisations Time series, Stat, Gauge, Bar gauge, Table, State timeline, Geomap
- Régler unités, calculs, seuils et correspondances de valeurs pour une lecture immédiate
- Surcharger l'affichage d'une série précise (override)
- Obtenir un tableau de bord du NOC lisible à 3 mètres

Déroulé

- Séquence S5.1 — Les types de graphiques (40 min)
- Séquence S5.2 — Paramétrer les graphiques (50 min)
- Séquence S5.3 — Bilan du jour 1 (15 min)

Séquence S5.1

Les types de graphiques

40 min

1. Time series : l'évolution dans le temps
2. Stat, Gauge, Bar gauge : une valeur clé
3. Table : le détail ligne par ligne
4. La ligne Synthèse après réglages
5. State timeline et Geomap
6. Carte et disponibilité des sites
7. Coordonnées pour la Geomap

🕮 Time series : l'évolution dans le temps

La visualisation par défaut

`Time series = Séries + Axe du temps + Style (lignes, barres, points)`

- **Graph styles** : lignes, barres, points ; épaisseur, remplissage, interpolation
- **Legend** : liste ou tableau, valeurs (`Last` , `Max` , `Mean`) à côté du nom
- **Tooltip** : une série (**Single**) ou toutes (**All**) au survol
- **Axis** : libellé, échelle (linéaire, logarithmique), position

💡 Bonne pratique

légende `{{region}}` (Lab 04.1) plutôt que la liste d'étiquettes complète

🕮 Stat, Gauge, Bar gauge : une valeur clé

Réduire une série à un chiffre

```
Valeur affichée = Calcul (Last, Mean, Max) sur la plage de temps
```

VISUALISATION	FORME	USAGE TELCÉA
Stat	Chiffre + fond ou courbe	Sites en service : `12` sur fond vert
Gauge	Cadran avec seuils	CPU de la VM : 13 %
Bar gauge	Barres horizontales ou verticales	Réussite des appels par région

- **Color mode** (Stat) : fond, valeur ou aucun ; **Graph mode** : mini-courbe ou non
- Gauge (refondue en Grafana 13) : **Show thresholds** = arcs de couleur des seuils

🕮 Table : le détail ligne par ligne

Une ligne par série, une colonne par étiquette

- Requête en **Instant** + **Format** `Table` : valeur actuelle de chaque série
- Transformation **Organize fields by name** : masquer, renommer, réordonner
- **Cell type** : texte, fond coloré, jauge, lien (par colonne via override)
- Tri par clic sur l'en-tête ; filtre par colonne (**Column filter**)

⇄ Avant / Après

Avant: `Time`, `__name__`, `instance`, `job`, `region`, `site`, `techno`, `Value` (1 / 0)
Après: `Site`, `Région`, `Techno`, `État` (« En service » sur fond vert)

La ligne Synthèse après réglages

Chiffre clé, consignes, détail, navigation



State timeline et Geomap

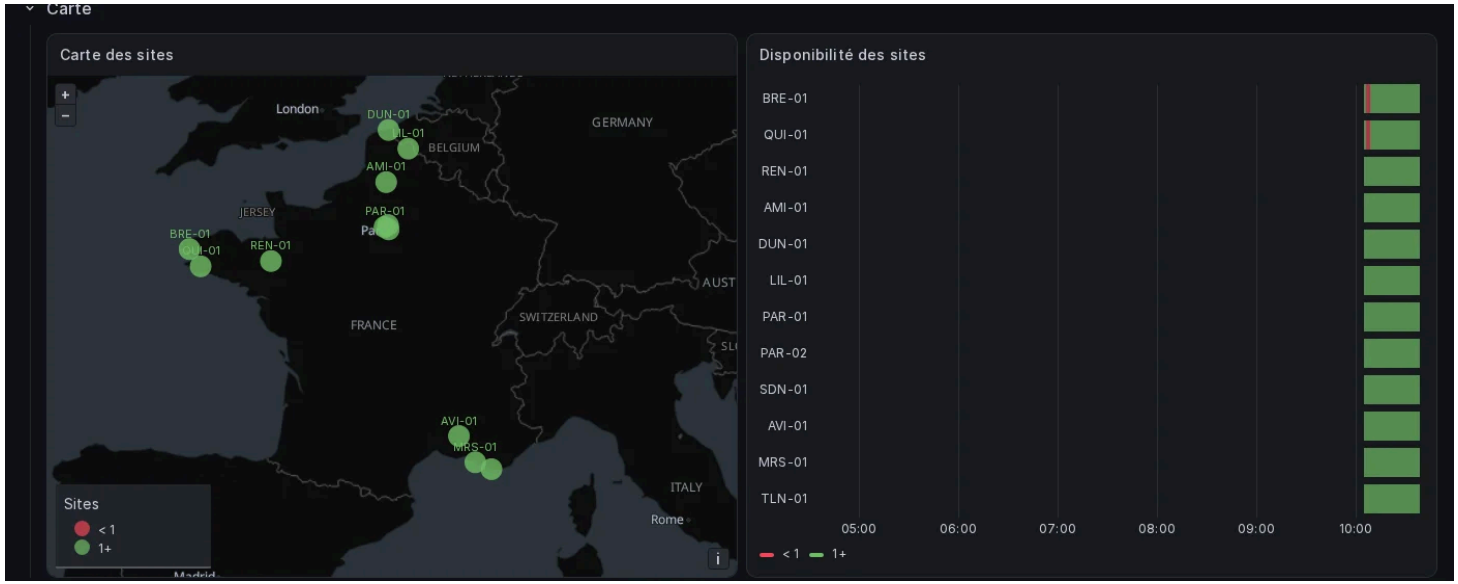
L'état dans le temps, l'état dans l'espace

VISUALISATION	DONNÉE ATTENDUE	USAGE TELCÉA
State timeline	Valeurs d'état (0 / 1, textes)	Disponibilité des 12 sites heure par heure
Status history	Valeurs à intervalle régulier	État horaire (cases colorées)
Geomap	Champs `latitude` et `longitude`	Carte des sites, couleur selon l'état

- Geomap : couche **Markers**, **Location** `Coords`, couleur selon un champ
- Fond de carte téléchargé depuis Internet (accès sortant requis)

🕮 Carte et disponibilité des sites

Où, et depuis quand



🕮 Coordonnées pour la Geomap

Joindre l'état et la description du site en PromQL

```promql
telcea_site_up * on(site) group_left(ville, latitude, longitude) telcea_site_info
```

- `telcea_site_info` : valeur toujours `1`, coordonnées portées par les étiquettes
- `on(site) group_left(...)` : recopie les étiquettes choisies sur chaque série d'état
- Étiquettes = texte → transformation **Convert field type** vers `Number`

⚠ Piège courant

coordonnées laissées en texte → aucun marqueur sur la carte

Séquence S5.2

Paramétrer les graphiques

50 min

1. Trois niveaux de réglages
2. Calculs : réduire une série
3. Unités
4. Seuils
5. Value mappings
6. Overrides
7. Transport et système après réglages

🕮 Trois niveaux de réglages

Du panel entier à une seule série

```
Affichage = Options du panel + Options standard + Overrides
```


- Options du panel : propres à la visualisation (légende, style, orientation)
- Options standard : tous les champs (unité, décimales, seuils, mappings)
- Override : exception pour un champ (nom, regex, requête)

🕮 Calculs : réduire une série

Option **Calculation** de Stat, Gauge, Bar gauge, légendes

CALCUL	RÉSULTAT	USAGE
Last *	Dernière valeur non nulle	État actuel (par défaut)
Mean	Moyenne sur la plage	Charge moyenne de la journée
Max / Min	Extrêmes	Pic de trafic
Total	Somme des points	Rarement pertinent sur un débit
Delta	Écart début / fin	Croissance d'un compteur

⚠ Piège courant

`Mean` sur `telcea_site_up` → `11.9` au lieu de `12`

🕮 Unités

Standard options → Unit : Grafana formate et convertit

UNITÉ	VALEUR BRUTE	AFFICHAGE
`bits/sec(SI)` (`bps`)	`776475341`	`776 Mb/s`
`Percent (0.0-1.0)` (`percentunit`)	`0.41`	`41%`
`Percent (0-100)` (`percent`)	`13.1`	`13.1%`
`bytes(IEC)`	`1073741824`	`1 GiB`
`seconds (s)`	`0.059`	`59 ms`

⚠ Piège courant

`percent` sur un ratio 0-1 → `0.41%` au lieu de `41%`

Seuils

Transformer une valeur en état : vert, orange, rouge

PANEL	SEUILS (MODE ABSOLUTE)	EFFET
Sites en service	rouge Base · orange 10 · vert 12	Fond rouge dès 9 sites
Liens de collecte	vert Base · orange 0.8 · rouge 0.9	Lignes pointillées à 80 % et 90 %
CPU de la VM	vert Base · orange 65 · rouge 90	Arcs de couleur de la jauge

- Mode **Percentage** : seuil relatif au min / max du champ
- **Show thresholds** (Time series) : lignes, zones ou lignes pointillées

🕮 Value mappings

Remplacer une valeur par un texte et une couleur

Mapping = Valeur (ou plage, regex, nul) → Texte + Couleur

TYPE	EXEMPLE TELCÉA	AFFICHAGE
Value	`1`	« En service » (vert)
Value	`0`	« Hors service » (rouge)
Range	`0.9` à `1`	« Saturé » (rouge)
Special	`Null` / `NaN`	« Pas de données » (gris)

- Même mapping sur Table, State timeline et Geomap : vocabulaire unique du NOC

🕮 Overrides

Une exception pour une série, sans toucher aux autres

SÉLECTEUR	EXEMPLE	PROPRIÉTÉ MODIFIÉE
Fields with name	`REN-PAR`	Couleur fixe violette (lien breton)
Fields with name	`TRANSMIT`	**Transform** : `Negative Y`
Fields with name	`État` (table)	**Cell type** : fond coloré
Fields with name matching regex	`PAR-.*`	Couleur commune aux sites parisiens

- Bouton **+ Add field override** en bas des options du panel

Transport et système après réglages

Unités, seuils en pointillés, jauges, émission en négatif



Séquence S5.3

Bilan du jour 1

15 min

1 Le jour 1 en un schéma

🕮 Le jour 1 en un schéma

De la stack vide au tableau de bord du NOC

- QCM d'ancrage du M05 = bilan du J1 (questions sur M03 à M05)

À retenir

Points clés

- Stat / Gauge : un chiffre + un calcul (`Last *` par défaut) ; Table : détail en **Instant**
- Unité adaptée (`bps` , `percentunit`) : Grafana convertit et formate
- Seuils = états colorés ; value mappings = vocabulaire du NOC (« En service »)
- Override = exception pour une série (couleur, axe négatif, type de cellule)

QCM d'ancrage (1/4)

Questions 1 à 2

1. Quelle requête PromQL donne le débit d'un compteur d'octets ?

- **A.** `telcea_cell_traffic_bytes_total`
- **B.** `rate(telcea_cell_traffic_bytes_total[5m])`
- **C.** `sum(telcea_cell_traffic_bytes_total)`
- **D.** `max(telcea_cell_traffic_bytes_total)`

Réponse B — `rate()` transforme le compteur d'octets en débit par seconde.

Révéler la réponse

2. Un ratio vaut `0.41` : quelle unité affiche `41%` ?

- **A.** `percentunit`
- **B.** `percent`
- **C.** `short`
- **D.** `bps`

Réponse A — `percentunit` attend un ratio 0-1.

Révéler la réponse

QCM d'ancrage (2/4)

Questions 3 à 4

3. Un Stat affiche `11.9` sites en service : quelle est la cause probable ?

- **A.** Unité incorrecte
- **B.** Seuil mal réglé
- **C.** Source de données en panne
- **D.** Calcul `Mean` au lieu de `Last *`

Réponse D — `Mean` moyenne les valeurs de la plage ; `Last *` donne l'état actuel.

Révéler la réponse

4. Comment afficher « En service » au lieu de `1` ?

- **A.** Une transformation Rename
- **B.** Un seuil à 1
- **C.** Un value mapping
- **D.** Une unité personnalisée

Réponse C — Un value mapping associe une valeur à un texte et une couleur.

Révéler la réponse

QCM d'ancrage (3/4)

Questions 5 à 6

5. Comment colorer uniquement la série `REN-PAR` en violet ?

- **A.** Modifier la couleur de toutes les séries
- **B.** Un override **Fields with name** `REN-PAR`
- **C.** Renommer la série
- **D.** Un seuil à 0.8

Réponse B — Override **Fields with name** : exception pour une série.

Révéler la réponse

6. Que faut-il à la Geomap pour placer des marqueurs ?

- **A.** Une étiquette `site`
- **B.** Une requête SQL
- **C.** Des champs numériques `latitude` et `longitude`
- **D.** Un plugin payant

Réponse C — La couche Markers attend des champs numériques de latitude et de longitude.

Révéler la réponse

QCM d'ancrage (4/4)

Questions 7 à 8

7. Où se trouvent la description et les tags d'un tableau de bord ?

A. Settings → General

B. Administration → Settings

C. Explore

D. Options du panel

Réponse A — Settings → General (M04).

Révéler la réponse

8. Pourquoi l'URL de la source Prometheus est-elle `http://prometheus:9090` ?

A. Le navigateur se connecte directement à Prometheus

B. Prometheus refuse `localhost`

C. Le port 9090 est réservé à Grafana

D. Le serveur Grafana résout le nom du service dans le réseau Docker

Réponse D — L'URL est résolue par le serveur Grafana dans le réseau Docker (M03).

Révéler la réponse

🎓 Module 6 : Tableaux de bord paramétrables

Durée
1 h (S6.1 : 40 min · S6.2 : 20 min)

Prérequis
M05

Fil rouge
« [TLC] Réseau — Vue nationale » lisible, aucune variable

Objectifs

- Créer des variables de type query, custom et interval, puis les utiliser dans les requêtes et les titres
- Chaîner deux variables (le site dépend de la région)
- Répéter un panel pour chaque valeur d'une variable
- Ranger les tableaux de bord : dossiers imbriqués, tags, favoris, restauration après suppression

Déroulé

- Séquence S6.1 — Les variables (40 min)
- Séquence S6.2 — Répéter et catégoriser (20 min)

Séquence S6.1

Les variables

40 min

1. Pourquoi des variables ?
2. Les types de variables
3. Variable de type Query
4. Utiliser une variable
5. Variables chaînées
6. Le tableau de bord régional
7. Pièges des variables

❓ Pourquoi des variables ?

Un tableau de bord pour 12 sites, pas 12 tableaux de bord

`Variable = Nom + Liste de valeurs → $nom remplacé dans les requêtes`

🔄 Avant / Après

Avant: 4 copies du tableau de bord (une par région) → 4 corrections à chaque évolution
Après: 1 tableau de bord + liste déroulante `Région` → 1 seule maintenance

- Liste en haut du tableau de bord ; sélection reportée dans l'URL (`?var-region=Bretagne`)
- Utilisable dans les requêtes, les titres de panels, les liens

💡 Bonne pratique

variables plutôt que copies (règle du support historique)

🕮 Les types de variables

Settings → Variables → **New variable**

TYPE	VALEURS	EXEMPLE TELCÉA
Query	Lues dans la source	`region` : 4 régions lues dans Prometheus
Custom	Liste saisie à la main	`techno` : `4G,5G`
Interval	Durées	`fenetre` : `1m,5m,15m,1h`
Constant	Valeur fixe, masquée	Seuil commun à plusieurs panels
Text box	Saisie libre	Recherche d'un code d'alarme
Data source	Sources d'un type	Basculer entre deux Prometheus

🕮 Variable de type Query

Liste dynamique : un nouveau site apparaît sans modification

CHAMP	VALEUR
Name / Label	`region` / `Région`
Data source	`Prometheus`
Query type	**Label values** (étiquette `region`, métrique `telcea_site_info`)
Multi-value / Include All value	Activés
Refresh	**On dashboard load**

Équivalent en requête classique :

```promql
label_values(telcea_site_info, region)
```

📖 Utiliser une variable

`$nom` dans la requête, le titre, la légende

```promql
sum by (site) (telcea_cell_users{region=~"$region", site=~"$site", techno=~"$techno"})
sum by (site) (rate(telcea_cell_traffic_bytes_total{direction="down"}[$fenetre])) * 8
```

- `=~` obligatoire en multi-valeur : `$region` devient `(Bretagne|Provence)`
- Titre `Débit par site ($fenetre)` → « Débit par site (15m) »

⚠ Piège courant

`region="$region"` (égalité) → « No data » dès 2 valeurs choisies

📖 Variables chaînées

La liste des sites dépend de la région choisie

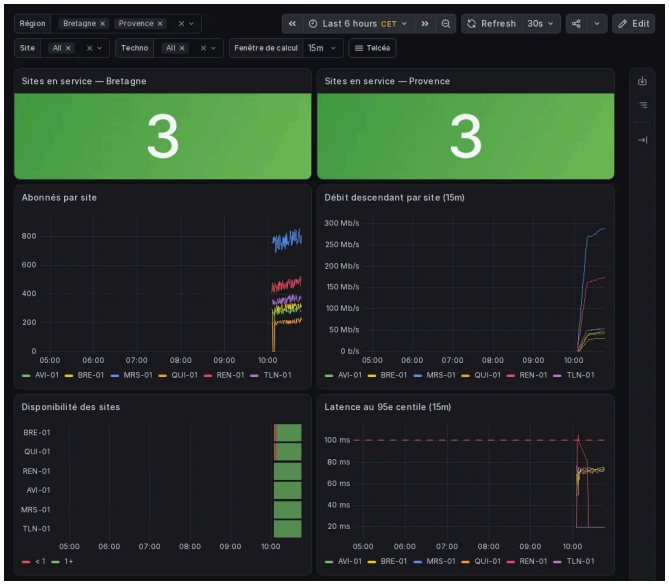
Variable fille = Requête filtrée par $variable_mère

```promql
label_values(telcea_site_info{region=~"$region"}, site)
```

- Ordre des variables = ordre d'évaluation (la mère AVANT la fille)

Le tableau de bord régional

Quatre variables, une seule maintenance



⚠ Pièges des variables

Une variable ne filtre que les séries qui portent l'étiquette

SYMPTÔME	CAUSE	CORRECTION
« No data » avec 2 valeurs	`=` au lieu de `=~`	`region=~"$region"`
« No data » sur la latence	Filtre `techno` : étiquette absente de la métrique	Retirer `techno` de cette requête
Liste des sites figée	**Refresh** sur `Never`	**On dashboard load**
Sites d'une autre région	Variable fille non filtrée	`{region=~"$region"}` dans sa requête

Séquence S6.2

Répéter et catégoriser

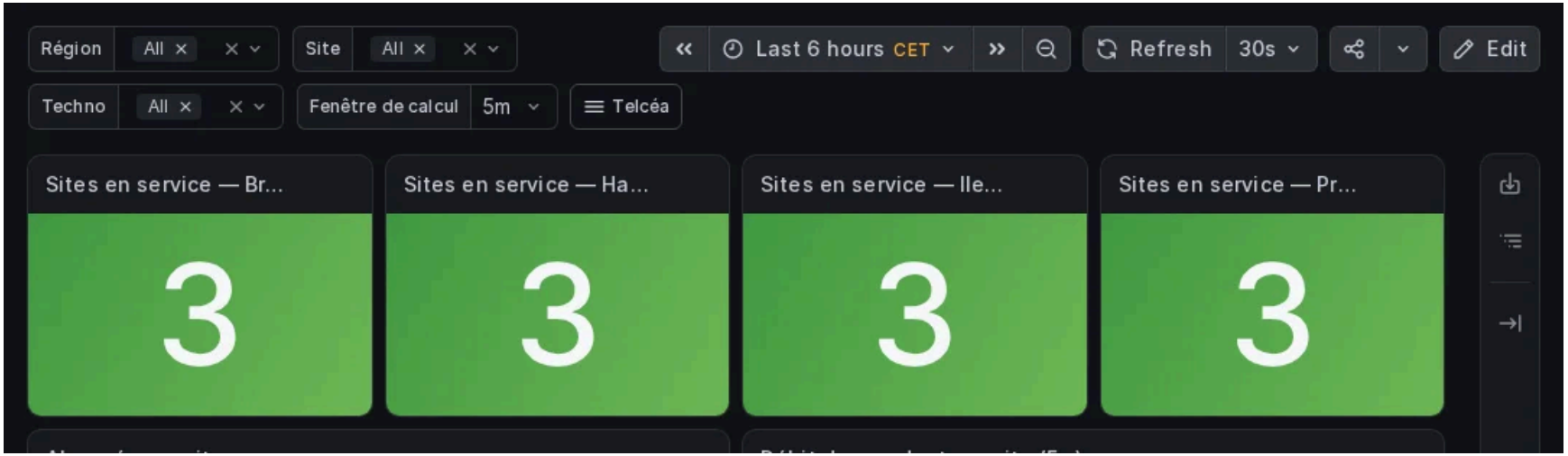
20 min

1. Panels répétés
2. Catégoriser les tableaux de bord
3. Restaurer un tableau de bord supprimé

🕮 Panels répétés

Un panel par valeur de la variable, généré automatiquement

`Repeat = Panel modèle × Valeurs sélectionnées`



- Options du panel → **Repeat options** → **Repeat by variable** : `region`
- Direction horizontale, **Max per row** : 4 ; titre `Sites en service — $region`

🕮 Catégoriser les tableaux de bord

Retrouver, trier, regrouper

OUTIL	USAGE TELCÉA
Dossiers imbriqués	`NOC Telcéa` → `Radio`, `Transport` (droits au M10)
Tags	`telcea`, `reseau`, `radio` : filtre et liens dynamiques
Favoris (étoile)	Écrans personnels de chaque technicien
Playlists	Rotation automatique sur l'écran mural
Liens de tableau de bord	Menu « Telcéa » : tous les tableaux de bord tagués

🕮 Restaurer un tableau de bord supprimé

Nouveauté Grafana 13 : la corbeille

`Suppression → Dashboards → Recently deleted → Restore`

- Liste : nom, date et auteur de la suppression, dossier d'origine
- Restauration dans un dossier au choix (le dossier d'origine a pu disparaître)
- Versions précédentes d'un tableau de bord : Settings → **Versions** (comparaison, restauration)

En entreprise

suppression par erreur d'un écran de NOC = restauration en 1 minute

À retenir

Points clés

- `$variable` dans les requêtes, titres, légendes ; `=~` en multi-valeur
- Query (lue dans la source), Custom (liste), Interval (fenêtre de `rate`)
- Chaînage : la variable fille filtre sur la mère ; ordre des variables important
- Repeat = un panel par valeur ; dossiers imbriqués, tags, corbeille pour organiser

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quel type de variable lit ses valeurs dans Prometheus ?

- **A.** Custom
- **B.** Constant
- **C.** Query
- **D.** Text box

Réponse C — Query : valeurs lues dans la source.

Révéler la réponse

2. Quel filtre accepte plusieurs régions sélectionnées ?

- **A.** `region="$region"`
- **B.** `region=~"$region"`
- **C.** `region!="$region"`
- **D.** `region:$region`

Réponse B — `=~` : la sélection multiple devient une expression régulière `(A|B)`.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Comment chaîner la variable `site` à la variable `region` ?

A. `label_values(site)`

B. Créer `site` avant `region`

C. `label_values(telcea_site_info{region=~"$region"}, site)`

D. Une variable Constant

Réponse C — La requête de la fille filtre sur la valeur de la mère.

Révéler la réponse

4. Pourquoi la latence affiche « No data » avec le filtre `techno=~"$techno"` ?

A. La métrique de latence ne porte pas l'étiquette `techno`

B. La latence n'est jamais collectée

C. La variable `techno` est mal nommée

D. Le panel est en Instant

Réponse A — Un filtre sur une étiquette absente exclut toutes les séries.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Quel réglage produit un Stat par région sélectionnée ?

- **A.** Une transformation
- **B.** Un override
- **C.** Un lien de tableau de bord
- **D.** **Repeat by variable** sur `region`

Réponse D — **Repeat by variable** : une copie du panel par valeur sélectionnée.

Révéler la réponse

6. Où restaurer un tableau de bord supprimé par erreur (Grafana 13) ?

- **A.** Administration → Settings
- **B.** Dashboards → Recently deleted
- **C.** Explore
- **D.** Impossible

Réponse B — Grafana 13 : Dashboards → Recently deleted → Restore.

Révéler la réponse

🎓 Module 7 : Utilisation des plugins

Durée
1 h (S7.1 : 20 min · S7.2 : 40 min)

Prérequis
M06

Fil rouge
tableaux de bord national et régional, visualisations natives uniquement

Objectifs

- Distinguer les types de plugins et leurs niveaux de signature
- Installer un plugin par l'interface, par `grafana cli` et par configuration
- Évaluer un plugin (signature, maintenance, compatibilité, support) avant installation
- Configurer Status Panel, D3 Gauge, Flow et Polystat pour l'écran mural du NOC

Déroulé

- Séquence S7.1 — Catalogue, signatures, installation (20 min)
- Séquence S7.2 — Les plugins de l'écran mural (40 min)

Séquence S7.1

Catalogue, signatures, installation

20 min

1. Un plugin ?
2. Le catalogue dans Grafana
3. Signatures
4. Installer un plugin
5. Compatibilité : la fin d'Angular
6. Évaluer avant d'installer

? Un plugin ?

Étendre Grafana sans modifier son code

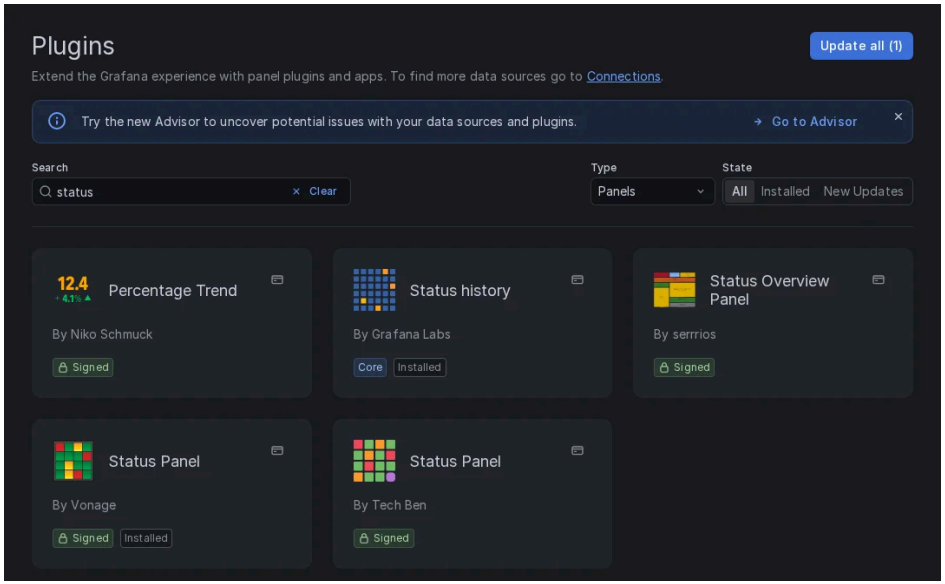
`Plugin = Panel | Source de données | Application`

TYPE	APPORTE	EXEMPLE
Panel	Une visualisation	Flow (schéma SVG animé)
Source de données	Un connecteur	Zabbix, MongoDB
Application	Pages + panels + sources	Drilldown, Advisor

- Catalogue : `grafana.com/grafana/plugins` et **Administration → Plugins and data → Plugins**
- Plugins internes (*Core*) : déjà présents (Time series, Prometheus, InfluxDB)

🕮 Le catalogue dans Grafana

Recherche, type, état (installé, mise à jour disponible)



🕮 Signatures

Qui garantit le code exécuté dans le navigateur du NOC ?

Signature = Éditeur identifié + Code non modifié depuis la publication

NIVEAU	SIGNÉ PAR	EXEMPLE
Grafana	Grafana Labs	Status Panel (Vonage), Polystat
Commercial	Éditeur partenaire	Plugins Enterprise
Community	Développeur indépendant	D3 Gauge, Flow
Private	Usage interne d'une organisation	Plugin maison
Non signé	Personne	Refusé par défaut

🕮 Installer un plugin

Trois méthodes, même résultat

MÉTHODE	COMMANDE OU ACTION	REDÉMARRAGE
Interface	Fiche du plugin → **Install**	Non
CLI (Grafana 13)	`grafana cli plugins install <ID> [<VERSION>]`	Oui
Configuration	`GF_PLUGINS_PREINSTALL_SYNC=<ID>@<VERSION>,...`	Au démarrage

BASH

```
docker compose exec grafana grafana cli plugins install andrewbmchugh-flow-panel
```

```
✔ Downloaded and extracted andrewbmchugh-flow-panel v1.20.1 zip successfully to /var/lib/grafana/plugins/andrewbmchugh-flow-panel
```

🕮 Compatibilité : la fin d'Angular

Grafana 12 a retiré les plugins écrits en AngularJS

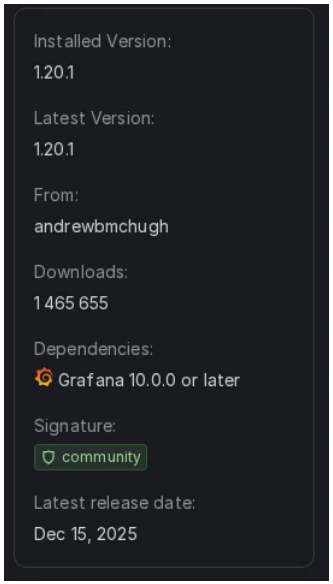
PLUGIN DU PROGRAMME	ÉTAT (NOVEMBRE 2026)	CHOIX DE LA FORMATION
Boom Table	Déprécié, Angular : ne se charge plus	Table native + transformations, ou Polystat
Diagram	Sans mise à jour depuis 04/2024	Flow (schéma SVG, mis à jour en 12/2025)
Status Panel (Vonage)	2.0.4, React, signé Grafana	Conservé
D3 Gauge	2.2.0 (04/2026), React	Conservé

⚠ Piège courant

migration vers 12 → « Panel plugin not found: yesoreyeram-boomtable-panel »

🕮 Évaluer avant d'installer

Fiche du plugin : version, dépendances, signature, date



- Grille : signature · dernière mise à jour · compatibilité · licence · dépôt et tickets · besoin couvert
- Un plugin = du code tiers dans le navigateur de tous les utilisateurs

Séquence S7.2

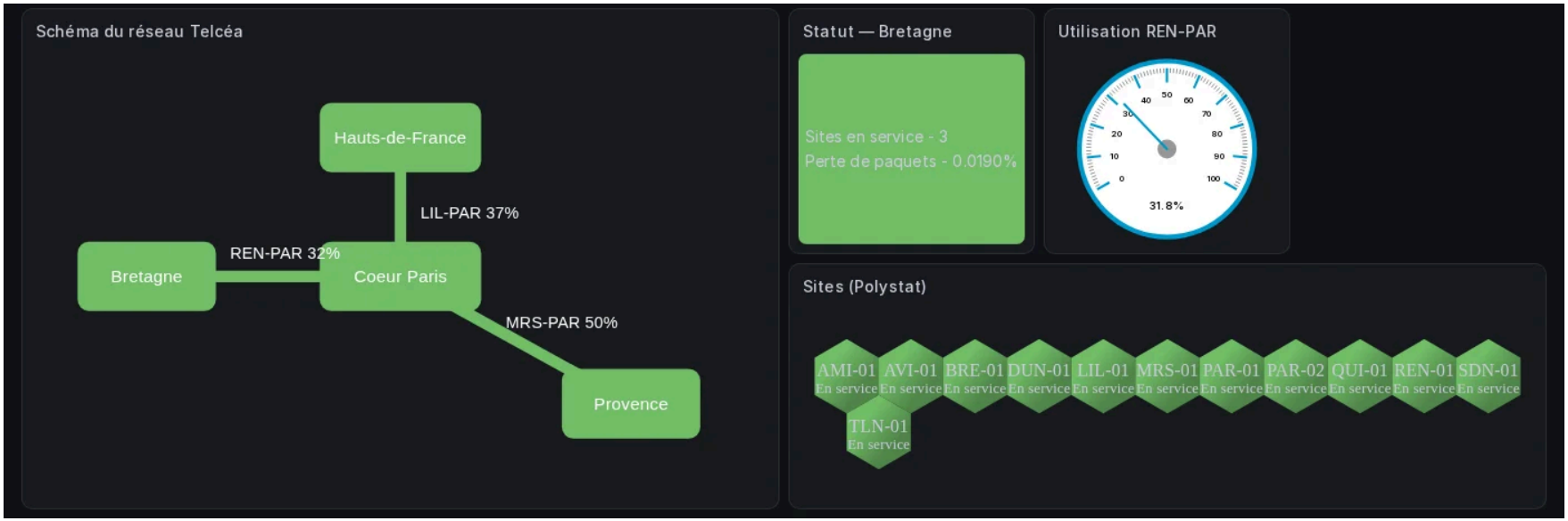
Les plugins de l'écran mural

40 min

1. L'écran mural du NOC
2. Status Panel
3. D3 Gauge et Polystat
4. Flow : le schéma du réseau
5. Flow : la configuration

L'écran mural du NOC

Quatre plugins, quatre usages



Status Panel

Un pavé de couleur pour plusieurs indicateurs

Couleur du pavé = Pire état parmi les indicateurs (OK, Warning, Critical)

- Une requête par indicateur ; nom affiché = **Display name** (override par requête)
- Seuils par indicateur : **Number Threshold**, valeurs **Warning** et **Critical**
- Sens détecté automatiquement : `Critical` < `Warning` → les grandes valeurs sont bonnes
- Bretagne : sites en service (warning 2, critique 1), perte de paquets (1 %, 2 %)

Bonne pratique

un Status Panel par région (panel répété sur `$region`)

🕮 D3 Gauge et Polystat

Jauge à aiguille, grille d'états

PLUGIN	FORME	RÉGLAGES CLÉS
D3 Gauge 2.2.0	Cadran à aiguille, graduations	Unité, min / max, bandes de seuils
Polystat 2.1.16	Hexagones, un par série	Calcul (**current**), seuils, mappings, colonnes

- Polystat : remplaçant naturel de Boom Table pour une vue « un pavé par site »
- Jauge native refondue en Grafana 13 : D3 Gauge seulement pour le rendu « cadran »

🕮 Flow : le schéma du réseau

Un dessin SVG animé par les métriques

Flow = SVG (identifiants) + YAML (identifiant → série → seuils)

- Identifiant SVG `cell-REN-PAR` ↔ série nommée `REN-PAR` (légende `{{link}}`)
- `strokeColor` (traits), `fillColor` (formes), `label` (texte + valeur)

🕮 Flow : la configuration

Extrait de `flow-config.yaml`

```yaml
cellIdPreamble: "cell-"
cells:
  REN-PAR:
    dataRef: "REN-PAR"
    strokeColor:
      thresholds:
        - {color: "green", level: 0}
        - {color: "orange", level: 0.8}
        - {color: "red", level: 0.9}
  Bretagne: {dataRef: "Bretagne", fillColor: {thresholds: *seuils-sites}}
```

⚠ Piège courant

`dataRef` ≠ nom de la série (légende absente) → élément gris

À retenir

Points clés

- `Plugin = Panel | Source | Application` ; signature = éditeur + intégrité
- Grafana 13 : `grafana cli plugins install <ID>` puis redémarrage, ou installation depuis l'interface
- Angular retiré (Grafana 12) : Boom Table → Polystat ou Table ; Diagram → Flow
- Évaluer AVANT d'installer : signature, date, compatibilité, licence, support

QCM d'ancrage (1/3)

Questions 1 à 2

1. Que garantit la signature d'un plugin ?

- **A.** L'identité de l'éditeur et l'intégrité du code publié
- **B.** Qu'il est gratuit
- **C.** Qu'il est compatible avec toutes les versions
- **D.** Qu'il ne contient aucun bogue

Réponse A — Signature : éditeur identifié et code identique à la version publiée.

Révéler la réponse

2. Quelle commande installe un plugin en Grafana 13 ?

- **A.** `grafana-cli plugins install <ID>`
- **B.** `apt install <ID>`
- **C.** `docker plugin install <ID>`
- **D.** `grafana cli plugins install <ID>`

Réponse D — `grafana-cli` supprimé en Grafana 13.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Pourquoi Boom Table ne fonctionne-t-il plus ?

- **A.** Il est devenu payant
- **B.** Il est écrit en AngularJS, retiré en Grafana 12
- **C.** Il nécessite InfluxDB
- **D.** Il n'a jamais existé

Réponse B — Plugins Angular retirés en Grafana 12.

Révéler la réponse

4. Après `grafana cli plugins install`, que faut-il faire ?

- **A.** Redémarrer Grafana
- **B.** Rien
- **C.** Recréer les sources de données
- **D.** Vider le cache du navigateur

Réponse A — Plugins installés par la CLI chargés au démarrage.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Un élément Flow reste gris : cause la plus probable ?

- **A.** Le SVG est trop grand
- **B.** Le plugin n'est pas signé
- **C.** Le tableau de bord n'est pas enregistré
- **D.** `dataRef` ne correspond à aucun nom de série

Réponse D — `dataRef` doit correspondre au nom d'une série.

Révéler la réponse

6. Quel critère n'aide PAS à évaluer un plugin ?

- **A.** Date de dernière mise à jour
- **B.** Niveau de signature
- **C.** Couleur de son icône
- **D.** Version minimale de Grafana requise

Réponse C — L'icône n'indique ni la qualité ni la maintenance.

Révéler la réponse

🎓 Module 8 : Partage du tableau de bord

Durée
45 min (S8.1 : 25 min · S8.2 : 20 min)

Prérequis
M07

Fil rouge
trois tableaux de bord du NOC (national, régional, écran mural)

Objectifs

- Partager un tableau de bord par lien interne (plage de temps figée), par lien externe et par snapshot
- Exporter un tableau de bord en JSON et importer celui d'un autre stagiaire
- Réutiliser un panel dans plusieurs tableaux de bord (library panel)
- Choisir le mode de partage adapté au destinataire (collègue, direction, prestataire)

Déroulé

- Séquence S8.1 — Partager un tableau de bord (25 min)
- Séquence S8.2 — Exporter et réutiliser (20 min)

Séquence S8.1

Partager un tableau de bord

25 min

1. Partager : à qui, et quoi ?
2. Lien interne et lien externe
3. Lien interne
4. Lien externe (ex-« public dashboard »)
5. Snapshot et export
6. Snapshot

❓ Partager : à qui, et quoi ?

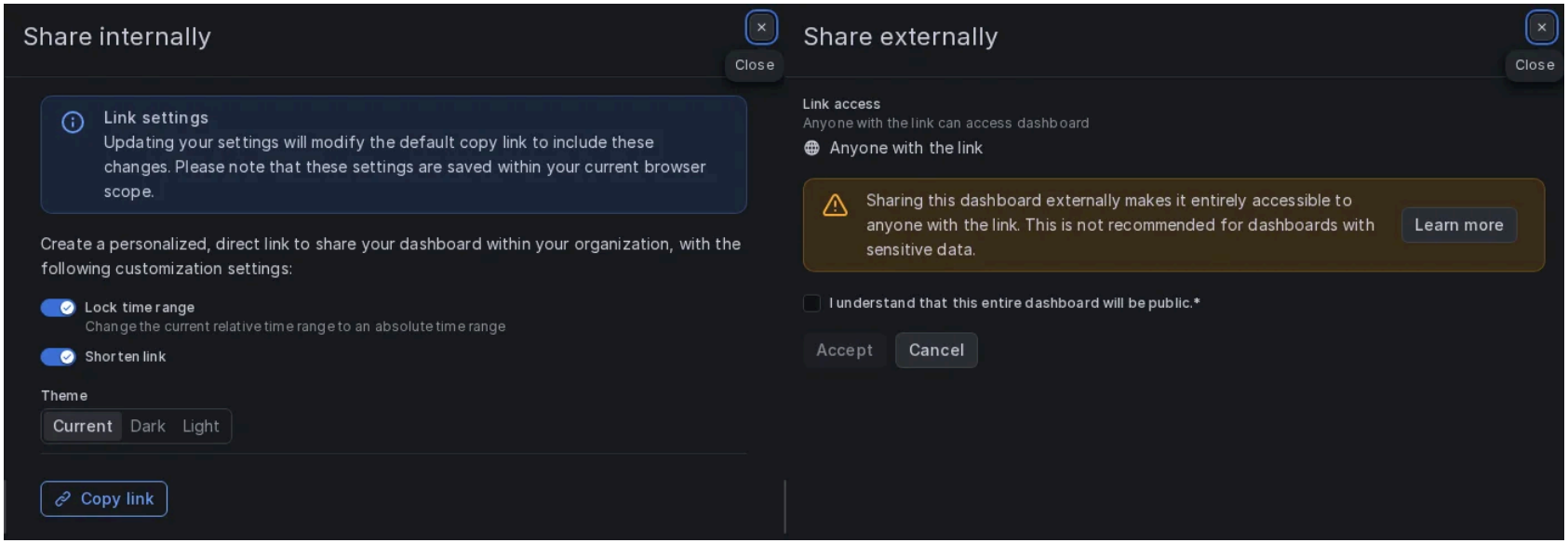
Le destinataire dicte le mode de partage

```
Mode de partage = Destinataire + Données exposées + Durée
```

DESTINATAIRE	BESOIN	MODE
Collègue du NOC (compte Grafana)	Même vue, même instant	Lien interne
Direction, sans compte	Vue en direct, lecture seule	Lien externe
Prestataire, après incident	Figer l'état, sans les requêtes	Snapshot
Autre instance Grafana	Reprendre le tableau de bord	Export JSON

🕮 Lien interne et lien externe

Bouton **Share** (barre du haut) → **Share internally** / **Share externally**



📖 Lien interne

Pour les utilisateurs de l'organisation, avec leurs droits

- **Lock time range** : `now-6h` devient une plage absolue (le destinataire voit le même instant)
- **Shorten link** : URL courte `/goto/...`
- **Theme** : sombre ou clair forcé
- Variables sélectionnées incluses dans le lien (`var-region=Bretagne`)

> **💡 Bonne pratique**
>
> lien à plage figée dans tout ticket d'incident (preuve horodatée)

🕮 Lien externe (ex-« public dashboard »)

Accès sans compte Grafana, en direct

```
Lien externe = Tableau de bord entier + Lecture seule + Sans authentification
```

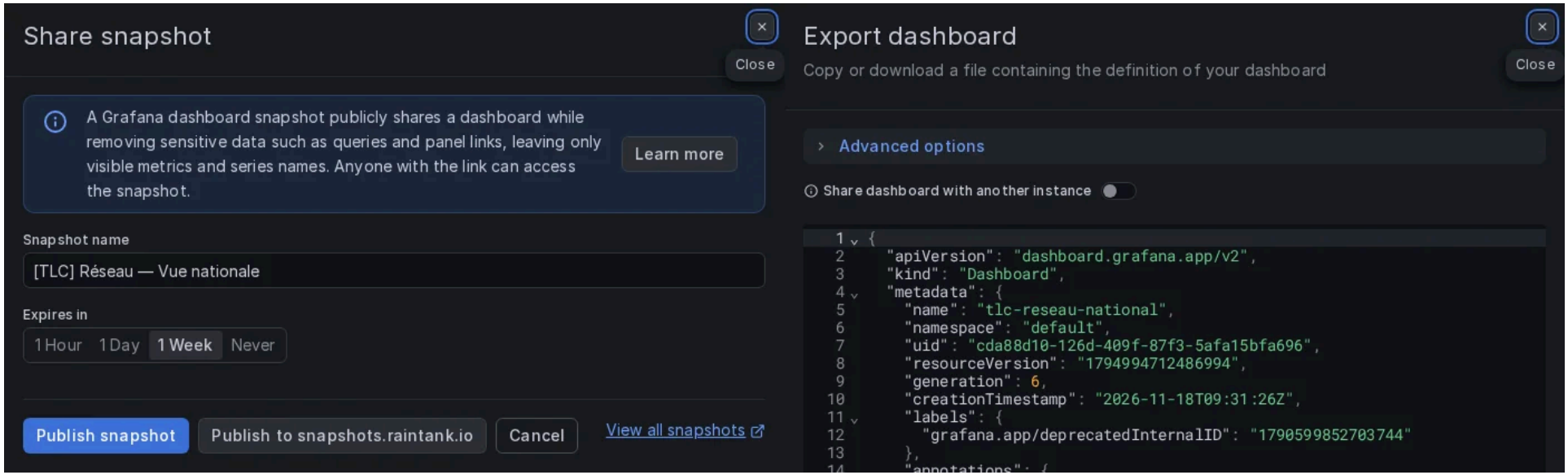
- **Anyone with the link** : toute personne disposant du lien (OSS)
- Case « I understand that this entire dashboard will be public » puis **Accept**
- **Pause access** : suspendre sans supprimer ; liste dans **Dashboards → Public dashboards**
- Partage à des personnes précises (**Only specific people**) : Enterprise et Cloud

⚠ Piège courant

variables et sélecteur de plage non modifiables par le destinataire

Snapshot et export

Figer les données, ou exporter la définition



Snapshot

Une photo du tableau de bord : données figées, requêtes retirées

`Snapshot = Données affichées + Mise en forme − Requêtes − Sources`

OPTION	VALEUR TELCÉA
Snapshot name	`[TLC] Réseau — incident tempête 18/11`
Expires in	`1 Hour`, `1 Day`, `1 Week`, `Never`
Publish snapshot	Stocké sur l'instance (**Snapshots**)
Publish to snapshots.raintank.io	Service externe (Grafana Labs)

- Publication externe = données de production hors du SI (à proscrire sauf accord)

Séquence S8.2

Exporter et réutiliser

20 min

1. Exporter et importer en JSON
2. Library panels
3. Ce que l'édition OSS ne fait pas

🕮 Exporter et importer en JSON

Copier un tableau de bord d'une instance à l'autre

- **Share dashboard with another instance** : sources remplacées par des paramètres d'import
- Import : fichier, JSON collé, ou identifiant d'un tableau de bord de `grafana.com`
- Grafana 13 : export au format v2 (`apiVersion: dashboard.grafana.app/v2`)

Library panels

Un panel, plusieurs tableaux de bord, une seule maintenance

`Library panel = Panel partagé ; modification → tous les tableaux de bord`

- Menu du panel → **More...** → **New library panel** (nom + dossier)
- Ajout : nouveau panel vide → **Use library panel**
- Liste et usages : **Dashboards** → **Library panels**
- **Unlink library panel** : copie locale indépendante

En entreprise

panel « Consignes d'astreinte » partagé par tous les écrans

🕮 Ce que l'édition OSS ne fait pas

Envoyer un tableau de bord par courriel : solutions selon l'édition

BESOIN	OSS (FORMATION)	ENTERPRISE / CLOUD
Rapport PDF planifié par courriel	Non	**Schedule a report**
Export PDF	Non	**Export as PDF**
Image d'un panel	Capture d'écran (Image Renderer retiré en 13)	Rendu serveur
Courriel avec un lien	Lien interne ou snapshot collé dans un courriel	Idem

- Courriels émis par Grafana OSS : alertes (M09) et invitations (M10)

À retenir

Points clés

- Lien interne : comptes Grafana, plage figée (**Lock time range**)
- Lien externe : sans compte, tableau de bord entier, en direct
- Snapshot : données figées, sans requêtes ; publication externe = sortie de données
- Export JSON (schéma v2) + import : copie entre instances ; library panel : réutilisation

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quel mode pour montrer l'état du réseau à la direction, sans compte Grafana, en direct ?

- **A.** Lien externe
- **B.** Snapshot
- **C.** Export JSON
- **D.** Library panel

Réponse A — Lien externe : en direct, sans compte.

Révéler la réponse

2. Quel réglage du lien interne garantit que le destinataire voit le même instant ?

- **A.** Shorten link
- **B.** Theme
- **C.** Pause access
- **D.** Lock time range

Réponse D — **Lock time range** convertit la plage relative en plage absolue.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Que contient un snapshot ?

- **A.** Les requêtes et les sources de données
- **B.** Uniquement les titres
- **C.** Les données affichées et la mise en forme, sans les requêtes
- **D.** Une image PNG

Réponse C — Snapshot : données affichées et mise en forme, sans requêtes ni sources.

Révéler la réponse

4. Quel risque présente Publish to snapshots.raintank.io ?

- **A.** Aucun
- **B.** Les données quittent l'infrastructure de l'entreprise
- **C.** Le tableau de bord est supprimé
- **D.** Les alertes sont désactivées

Réponse B — Publication sur un service public de Grafana Labs : sortie des données.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Pourquoi activer Share dashboard with another instance à l'export ?

- **A.** Pour chiffrer le fichier
- **B.** Pour réduire la taille du fichier
- **C.** Pour exporter en PDF
- **D.** Pour remplacer les sources par des paramètres choisis à l'import

Réponse D — Sources remplacées par des paramètres d'import.

Révéler la réponse

6. Un library panel est modifié : quel effet ?

- **A.** Modification visible dans tous les tableaux de bord qui l'utilisent
- **B.** Aucun sur les autres tableaux de bord
- **C.** Les autres copies sont supprimées
- **D.** Le panel devient public

Réponse A — Un library panel est partagé : toute modification est visible partout.

Révéler la réponse

Module 9 : Gestion des alertes

Durée
1 h 30 (S9.1 : 45 min · S9.2 : 45 min)

Prérequis
M05 (seuils), M06

Fil rouge
écrans du NOC complets, aucune alerte

Objectifs

- Créer une règle d'alerte Grafana (requête, condition, évaluation, labels, annotations)
- Router les notifications vers l'équipe concernée par une politique fondée sur les labels
- Suspendre les notifications pendant une intervention (silence)
- Situer l'Alertmanager externe de Prometheus par rapport à l'alerting de Grafana

Déroulé

- Séquence S9.1 — Les règles d'alerte (45 min)
- Séquence S9.2 — Notifier, router, suspendre (45 min)

Séquence S9.1

Les règles d'alerte

45 min

1. Une alerte, de la mesure au courriel
2. Deux façons de gérer les règles
3. Créer une règle
4. Requête et condition
5. Évaluation
6. Cycle de vie d'une instance
7. Labels et annotations
8. Les alertes du NOC

🕮 Une alerte, de la mesure au courriel

Grafana évalue, route, notifie

Alerte = Règle (requête + condition) → Instance → Politique → Point de contact

- Une règle produit une instance par série (un site hors service = une instance)
- Labels de la règle (`equipe`, `severite`) : clés du routage

🕮 Deux façons de gérer les règles

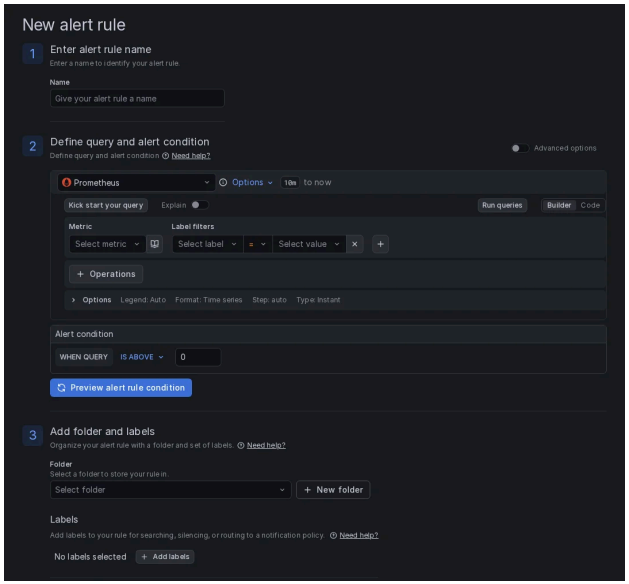
Grafana ou la source de données

GESTION	RÈGLES STOCKÉES DANS	USAGE
Grafana-managed (défaut)	Base de Grafana	Toutes sources, plusieurs requêtes, expressions
Data source-managed	Prometheus, Loki, Mimir	Règles existantes de Prometheus (lecture dans Grafana)

- Telcéa : règles Grafana pour le NOC ; règle Prometheus `SiteRadioHorsService` déjà en place (Alertmanager)
- Menu **Alerting** → **Alert rules** : les deux types dans une même liste

Créer une règle

Alerting → Alert rules → **New alert rule** : 6 sections



🕮 Requête et condition

Section 2 : que mesure-t-on, quand est-ce anormal ?

```promql
telcea_site_up
```

ÉLÉMENT	VALEUR TELCÉA
Requête A	`telcea_site_up` (Instant, Prometheus)
Condition (**Alert condition**)	WHEN QUERY **IS BELOW** `1`
Mode avancé	Expressions **Reduce**, **Math**, **Threshold** enchaînées

- **Preview alert rule condition** : état de chaque instance avant enregistrement

🕮 Évaluation

Section 4 : à quelle fréquence, après combien de temps ?

`Alerte = Condition vraie pendant la période d'attente (pending period)`

RÉGLAGE	VALEUR	EFFET
Evaluation group	`telcea-1m` (intervalle 1 min)	Règles évaluées ensemble
Pending period	`1m`	Pas d'alerte sur un pic d'une minute
Keep firing for	`0s` (None)	Retour immédiat à Normal
No data / Error	`NoData`, `Error`	État dédié si la requête échoue

🕮 Cycle de vie d'une instance

De Normal à Firing, et retour

- Notification envoyée au passage en **Alerting**, puis à la résolution
- **Recovering** : seulement si **Keep firing for** > 0

🕮 Labels et annotations

Labels pour router, annotations pour informer

TYPE	CLÉ	VALEUR TELCÉA
Label	`equipe`	`radio` ou `transport`
Label	`severite`	`critique`, `majeure`, `mineure`
Annotation	`summary`	`Site {{ $labels.site }} ({{ $labels.region }}) hors service`
Annotation	`runbook_url`	Procédure d'intervention

- `{{ $labels.site }}` : remplacé par la valeur de l'instance (BRE-01)
- Labels de la série (`site` , `region`) ajoutés automatiquement à l'instance

📖 Les alertes du NOC

Trois règles, deux équipes (liste de l'atelier historique)

RÈGLE	CONDITION	PENDING	LABELS
Site radio hors service	`telcea_site_up` < 1	1m	radio, critique
Lien de collecte saturé	utilisation > 0.8	2m	transport, majeure
Site sur batterie	`telcea_site_on_battery` > 0	1m	radio, mineure

🏢 En entreprise

seuil = niveau où l'action est nécessaire, pas niveau de panne déjà visible

Séquence S9.2

Notifier, router, suspendre

45 min

1. Points de contact
2. Politiques de notification
3. Silences
4. La tempête vue du NOC
5. Alertmanager externe

🕮 Points de contact

Où et comment notifier

`Point de contact = Intégration (courriel, Teams, Slack, webhook) + Paramètres`

POINT DE CONTACT	INTÉGRATION	DESTINATION (STACK)
NOC Telcéa	Email	`noc@telcea.example`
Équipe radio	Email	`radio@telcea.example`
Équipe transport	Email	`transport@telcea.example`
Alertmanager externe	Alertmanager	`http://alertmanager:9093`

- Bouton **Test** : notification factice ; courriels reçus dans Mailpit (`/mail/`)

🕮 Politiques de notification

Router selon les labels, de la racine vers les branches

- Alerte sans politique correspondante → politique par défaut (NOC)
- Regroupement : `group_by` (`alertname` , `region`), **Group wait** 30 s, **Repeat interval** 1 h
- **Mute timings** : pas de notification sur une plage horaire récurrente (nuit, week-end)

🕮 Silences

Suspendre les notifications le temps d'une intervention

`Silence = Matchers de labels + Début + Fin (+ commentaire)`

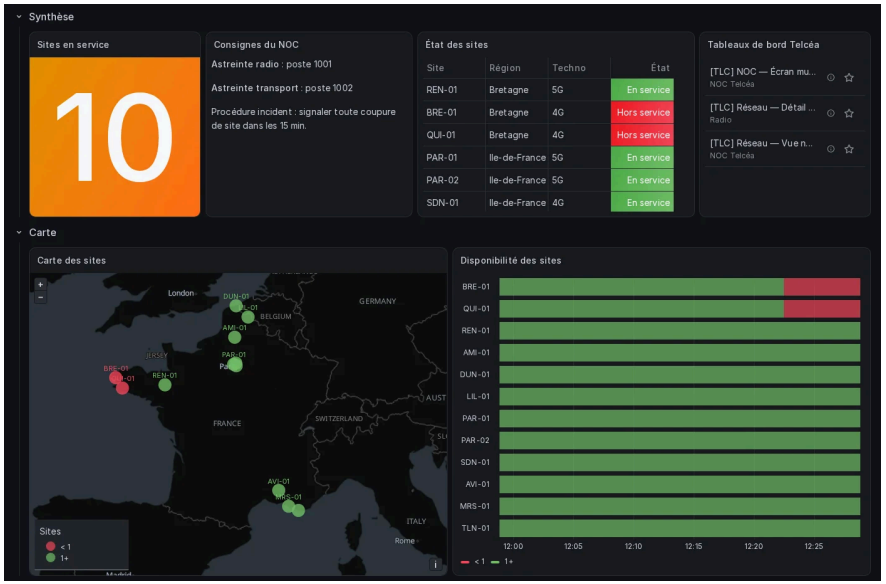
- Alerting → **Silences** → **Create silence** (ou bouton **Silence** sur une alerte)
- Exemple : `site = BRE-01` pendant 2 h (remplacement de l'alimentation)
- L'alerte reste évaluée et visible ; seule la notification est suspendue

⚠ Piège courant

silence sans date de fin réaliste → alerte réelle ignorée des jours plus tard

La tempête vue du NOC

Bretagne, 12 h 22 : deux sites coupés, alertes en 2 à 6 minutes



🕮 Alertmanager externe

L'outil d'alerte de l'écosystème Prometheus

- Quand : règles déjà écrites dans Prometheus, routage commun à plusieurs outils
- Grafana peut l'afficher (source **Alertmanager**) et lui transmettre ses alertes

À retenir

Points clés

- `Règle = Requête + Condition + Évaluation (pending) + Labels + Annotations`
- États : Normal → Pending → Alerting → (Recovering) → Normal
- Routage par labels : politique par défaut + branches `equipe = …`
- Silence = suspension ponctuelle ; mute timing = suspension récurrente

QCM d'ancrage (1/3)

Questions 1 à 2

1. Une règle porte le label `equipe=radio` : qui décide du destinataire ?

- **A.** La règle elle-même
- **B.** La politique de notification qui correspond au label
- **C.** Le tableau de bord
- **D.** Prometheus

Réponse B — La politique dont les matchers correspondent aux labels choisit le point de contact.

Révéler la réponse

2. À quoi sert la période d'attente (pending period) ?

- **A.** Retarder l'envoi de 24 h
- **B.** Répéter la notification
- **C.** Désactiver la règle la nuit
- **D.** Exiger que la condition reste vraie un certain temps avant l'alerte

Réponse D — Condition vraie pendant toute la période avant le passage en Alerting.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Combien d'instances produit la règle `telcea_site_up < 1` si 2 sites sont coupés ?

- **A.** 2
- **B.** 0
- **C.** 1
- **D.** 12

Réponse A — Une instance par série en anomalie.

Révéler la réponse

4. Une intervention est prévue sur BRE-01 de 14 h à 16 h : que créer ?

- **A.** Une nouvelle règle
- **B.** Un point de contact
- **C.** Un silence sur `site = BRE-01`
- **D.** Un snapshot

Réponse C — Silence sur `site = BRE-01` pour la durée de l'intervention.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Une alerte n'a aucun label `equipe` : où part la notification ?

- **A.** Nulle part
- **B.** Vers la politique par défaut (NOC)
- **C.** Vers toutes les équipes
- **D.** Vers Prometheus

Réponse B — Politique par défaut.

Révéler la réponse

6. Que permet l'annotation `{{ $labels.site }}` ?

- **A.** Filtrer les alertes
- **B.** Router vers le site
- **C.** Créer un silence
- **D.** Insérer le site concerné dans le message

Réponse D — Valeur du label de l'instance insérée dans le message.

Révéler la réponse

🎓 Module 10 : Utilisateurs, équipes et droits

Durée
45 min (S10.1 : 45 min)

Prérequis
M06 (dossiers), M09

Fil rouge
un seul compte `admin`, tout le monde peut tout modifier

Objectifs

- Créer des comptes et leur attribuer un rôle d'organisation adapté
- Regrouper les utilisateurs en équipes et donner des droits par dossier
- Appliquer le moindre privilège et le prouver par un test positif et un test négatif
- Situer les comptes de service et l'authentification externe (LDAP, SSO)

Déroulé

- Séquence S10.1 — Comptes, équipes, permissions (45 min)

Séquence S10.1

Comptes, équipes, permissions

45 min

1. Qui peut faire quoi ?
2. Rôles d'organisation
3. Créer des comptes
4. Équipes
5. Permissions sur un dossier
6. Moindre privilège : le prouver
7. Comptes de service
8. Organisations

❓ Qui peut faire quoi ?

Trois niveaux, du plus large au plus précis

Droit effectif = Rôle d'organisation + Permissions (dossier, tableau de bord) par utilisateur ou équipe

- Permissions héritées : dossier → sous-dossiers → tableaux de bord
- Administrateur du serveur (*Grafana server admin*) : au-dessus de toutes les organisations

📖 Rôles d'organisation

Le socle commun à tous les dossiers

RÔLE	PEUT	USAGE TELCÉA
Viewer	Consulter, explorer selon réglages	Direction, techniciens (base)
Editor	Créer et modifier tableaux de bord et alertes	Responsables d'équipe
Admin	Gérer utilisateurs, équipes, sources, plugins	Administrateur du NOC
No basic role	Rien sans permission explicite	Prestataires externes

- Rôle choisi à la création ou dans **Organization users** (**Change role**)

🕮 Créer des comptes

Administration → Users and access → Users

MÉTHODE	ACTION	RÉSULTAT
Création directe	**New user** : nom, courriel, login, mot de passe	Compte actif immédiatement
Invitation	**Invite** + **Send invite email**	Courriel avec lien d'activation (Mailpit)
Annuaire	LDAP, OAuth / SSO (Azure AD, Okta, Keycloak)	Comptes créés à la première connexion

- Comptes Telcéa : `tech.radio`, `tech.transport`, `directeur` (rôle Viewer)

En entreprise

SSO de l'entreprise et équipes synchronisées depuis l'annuaire

🕮 Équipes

Donner des droits à un groupe, pas à chaque personne

Équipe = Membres + Permissions (une seule fois)

- Administration → Users and access → **Teams** → **New team** : `Radio`, `Transport`
- Ajout des membres dans la fiche de l'équipe
- Arrivée d'un technicien : ajout dans l'équipe → tous les droits de l'équipe

⇄ Avant / Après

Avant: 12 techniciens × 3 dossiers = 36 permissions individuelles
Après: 2 équipes × 1 dossier = 2 permissions

🕮 Permissions sur un dossier

Dossier → **Folder actions** → **Manage permissions** → **Add a permission**

DOSSIER	BÉNÉFICIAIRE	PERMISSION
`NOC Telcéa`	Rôle Viewer / Editor (par défaut)	View / Edit
`NOC Telcéa / Radio`	Équipe Radio	Edit
`NOC Telcéa / Transport`	Équipe Transport	Edit
Tous	Rôle Admin	Admin

- Niveaux : **View** (lire), **Edit** (modifier), **Admin** (gérer les droits)

🕮 Moindre privilège : le prouver

Un test positif ET un test négatif par profil

PROFIL	ACTION	ATTENDU (BANC)
`tech.radio`	Modifier le tableau de bord de `Radio`	Autorisé (HTTP 200)
`tech.transport`	Modifier le tableau de bord de `Radio`	Refusé (HTTP 403)
`directeur`	Consulter le tableau de bord régional	Autorisé (HTTP 200)
`directeur`	Modifier ce tableau de bord	Refusé (HTTP 403)

⚠ Piège courant

test uniquement positif → droits trop larges jamais détectés

🕮 Comptes de service

Pour les scripts et les outils, pas pour les personnes

```
Compte de service = Rôle + Jetons (révocables, à durée limitée)
```

- Administration → Users and access → **Service accounts** → **Add service account**
- **Add service account token** : jeton affiché UNE seule fois
- Usage : sauvegarde des tableaux de bord, import automatique, outils de rattrapage

⚠ Piège courant

script qui utilise le mot de passe d'un administrateur humain

Organisations

Cloisonnement complet dans une même instance

- Une organisation = ses utilisateurs, sources, tableaux de bord, alertes
- Usage : clients ou filiales totalement séparés (pas de partage entre organisations)
- Telcéa : une seule organisation ; séparation par dossiers et équipes

Bonne pratique

dossiers + équipes avant de multiplier les organisations

À retenir

Points clés

- `Droit effectif = Rôle d'organisation + Permissions héritées des dossiers`
- Équipes = droits donnés une fois pour un groupe
- Moindre privilège : Viewer par défaut, Edit sur le dossier de son équipe
- Test positif + test négatif ; comptes de service pour les scripts

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quel rôle d'organisation pour un directeur qui consulte les écrans ?

- **A.** Admin
- **B.** Editor
- **C.** Viewer
- **D.** Grafana server admin

Réponse C — Viewer : consultation seule.

Révéler la réponse

2. Où donner à l'équipe Radio le droit de modifier ses tableaux de bord ?

- **A.** Dans chaque panel
- **B.** Dans les permissions du dossier Radio
- **C.** Dans Explore
- **D.** Dans les préférences de l'organisation

Réponse B — Permission Edit de l'équipe sur le dossier Radio.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Un technicien rejoint l'équipe Transport : que faire ?

- **A.** Recréer toutes les permissions
- **B.** Lui donner le rôle Admin
- **C.** Dupliquer les tableaux de bord
- **D.** L'ajouter comme membre de l'équipe Transport

Réponse D — Membre de l'équipe : hérite de ses permissions.

Révéler la réponse

4. Que prouve un test négatif ?

- **A.** Que l'utilisateur peut se connecter
- **B.** Que le tableau de bord s'affiche
- **C.** Qu'une action non autorisée est bien refusée
- **D.** Que l'équipe existe

Réponse C — Le refus attendu est vérifié.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Un script d'export nocturne doit appeler l'API : quel compte utiliser ?

- **A.** Un compte de service avec un jeton
- **B.** Le compte `admin`
- **C.** Le compte du directeur
- **D.** Un compte anonyme

Réponse A — Compte de service + jeton à droits limités.

Révéler la réponse

6. Quel niveau de permission permet de gérer les droits d'un dossier ?

- **A.** View
- **B.** Admin
- **C.** Edit
- **D.** Viewer

Réponse B — Admin sur un dossier : gestion de ses permissions.

Révéler la réponse

🎓 Module 11 : Croisement de données

Durée
40 min (S11.1 : 40 min)

Prérequis
M03 (sources Prometheus et InfluxDB), M05

Fil rouge
chaque panel interroge une seule source

Objectifs

- Afficher dans un même panel des requêtes de deux sources différentes (source Mixed)
- Joindre, calculer et réorganiser des résultats avec les transformations
- Construire un tableau de synthèse par site à partir de 4 requêtes
- Diagnostiquer un résultat vide après jointure (alignement temporel, noms de champs)

Déroulé

- Séquence S11.1 — Sources mixtes et transformations (40 min)

Séquence S11.1

Sources mixtes et transformations

40 min

1. Pourquoi croiser ?
2. La source Mixed
3. Mixed en pratique
4. Les transformations
5. Le tableau de bord de croisement
6. Synthèse par site : 4 requêtes, 1 tableau
7. Pièges du croisement

❓ Pourquoi croiser ?

Une question, plusieurs bases

`Croisement = Plusieurs requêtes (une ou plusieurs sources) + Transformations`

QUESTION DU NOC	SOURCE 1	SOURCE 2
Les deux collecteurs de la VM sont-ils d'accord ?	Prometheus (node-exporter)	InfluxDB (Telegraf)
Quel site a le débit par abonné le plus faible ?	Prometheus (abonnés)	Prometheus (trafic)
Le pic de CPU coïncide-t-il avec un pic de trafic ?	InfluxDB (CPU)	Prometheus (trafic)

- Reprise de l'exercice historique 7 (Prometheus + InfluxDB), en SQL au lieu de Flux

La source Mixed

Une requête par source, dans le même panel

- Sélecteur **Data source** du panel → `-- Mixed --` ; chaque requête choisit sa source
- Même plage de temps pour toutes les requêtes

Mixed en pratique

CPU de la VM vu par deux collecteurs

PROMQL

```
100 * (1 - avg(rate(node_cpu_seconds_total{mode="idle"}[1m])))   # A : node-exporter
```

SQL

```
SELECT $__dateBin(time) AS time, avg(100 - usage_idle) AS telegraf       -- B, InfluxDB
FROM cpu WHERE $__timeFilter(time) GROUP BY 1 ORDER BY 1
```

- Unité commune (`percent`) : les deux courbes sur le même axe
- **Query options** → **Min interval** `1m` : pas identique pour les deux sources

🕮 Les transformations

Retravailler les résultats AVANT la visualisation

```
Affichage = Requêtes → Transformation 1 → Transformation 2 → … → Visualisation
```

TRANSFORMATION	EFFET
Join by field	Aligne sur un champ commun (temps)
Merge series/tables	Fusionne des tables (même clé)
Add field from calculation	Opération entre deux champs
Organize fields by name	Masquer, renommer, ordonner
Filter fields by name	Garder ou exclure des champs
Group by / Sort by	Regrouper, agréger, trier

🕮 Le tableau de bord de croisement

Courbes croisées, écart calculé, synthèse par site



📌 Synthèse par site : 4 requêtes, 1 tableau

Le remplaçant de Boom Table, sans plugin

REQUÊTE (INSTANT, TABLE)	COLONNE FINALE
`sum by (site, region) (telcea_cell_users)`	Abonnés
`sum by (site) (rate(telcea_cell_traffic_bytes_total{direction="down"}[5m])) * 8`	Débit descendant
`histogram_quantile(0.95, sum by (le, site) (rate(...[5m])))`	Latence p95
`sum by (site) (telcea_site_up)`	État

- **Merge** (colonnes `Value #A` à `Value #D`) → **Add field from calculation** (débit / abonnés) → **Organize** → **Sort by**

⚠ Pièges du croisement

Une jointure qui ne trouve rien = panel vide

SYMPTÔME	CAUSE	CORRECTION
Écart vide (« No data »)	Horodatages différents (15 s contre 1 min)	Même intervalle : **Min interval** `1m`
Jointure sans effet	Champ `Time` d'un côté, `time` de l'autre	Jointure sur le champ temps par défaut
Lignes dupliquées par site	Étiquettes différentes (`region` d'un seul côté)	Mêmes étiquettes ou **Organize**
Calcul faux	Unités différentes (octets / bits)	Convertir dans la requête (`* 8`)

Bonne pratique

bouton **Table view** de l'éditeur pour voir le résultat de chaque étape

À retenir

Points clés

- Source **Mixed** : une requête par source dans un même panel
- Transformations enchaînées dans l'ordre : jointure, calcul, organisation, tri
- Même intervalle et même nom de champ de jointure, sinon résultat vide
- Table + Merge + calculs : synthèse par site sans plugin (remplace Boom Table)

QCM d'ancrage (1/3)

Questions 1 à 2

1. Comment interroger Prometheus et InfluxDB dans un même panel ?

- **A.** Impossible
- **B.** Deux panels superposés
- **C.** Source `-- Mixed --`, une source par requête
- **D.** Un plugin payant

Réponse C — Source Mixed : une source par requête.

Révéler la réponse

2. Quelle transformation crée une colonne « débit / abonnés » ?

- **A.** Add field from calculation
- **B.** Organize fields by name
- **C.** Filter fields by name
- **D.** Sort by

Réponse A — Add field from calculation, opération binaire.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Une jointure par le temps donne un panel vide : cause probable ?

- **A.** Plage de temps trop longue
- **B.** Horodatages des deux sources non alignés
- **C.** Unité incorrecte
- **D.** Source par défaut absente

Réponse B — Horodatages non alignés : aucune paire à joindre.

Révéler la réponse

4. Dans quel ordre les transformations s'appliquent-elles ?

- **A.** De haut en bas, chacune sur le résultat de la précédente
- **B.** Au hasard
- **C.** En parallèle
- **D.** Après la visualisation

Réponse A — Ordre de la liste, chaque étape sur le résultat précédent.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. Quelle transformation renomme `Value #A` en « Abonnés » ?

- **A.** Merge series/tables
- **B.** Group by
- **C.** Join by field
- **D.** Organize fields by name

Réponse D — Organize fields by name (renommage).

Révéler la réponse

6. Quel outil de l'éditeur montre le résultat intermédiaire des transformations ?

- **A.** Explore
- **B.** Advisor
- **C.** Query inspector / Table view
- **D.** Settings

Réponse C — Table view / Query inspector : données après transformations.

Révéler la réponse

🎓 Module 12 : Développement de plugin et bilan

Durée
25 min (S12.1 : 10 min · S12.2 : 15 min)

Prérequis
M07

Fil rouge
NOC Telcéa supervisé, alerté, organisé

Objectifs

- Décrire la marche à suivre pour créer un plugin de panel (outils, structure, test, signature)
- Lire le fichier `plugin.json` d'un plugin et en déduire son type, son identifiant et sa compatibilité
- Faire le bilan des 2 jours au regard des objectifs du programme
- Évaluer ses acquis (QCM final)

Déroulé

- Séquence S12.1 — Créer un plugin : la marche à suivre (10 min)
- Séquence S12.2 — Bilan et évaluation (15 min)

Séquence S12.1

Créer un plugin : la marche à suivre

10 min

1. De l'idée au catalogue
2. Générer le squelette
3. La structure d'un plugin de panel
4. Le cycle de développement

🕮 De l'idée au catalogue

Six étapes, outillées par Grafana Labs

- Prérequis : Node.js 22 ou plus (`@grafana/create-plugin` 9.0.4), Docker, Git
- Plugin interne : signature **private** (organisation) ; catalogue public : revue par Grafana Labs

🕮 Générer le squelette

Une commande, des questions (type, nom, organisation)

```bash
npx @grafana/create-plugin@latest \
  --plugin-type=panel --plugin-name=noc-status --org-name=telcea --no-backend
```

Sortie relevée au banc (extrait) :

```
✓ Scaffolded telcea-nocstatus-panel panel plugin
✓ Added basic E2E test (Playwright)
✓ Configured development environment (Docker)
```

- Types proposés : App, Data source, Panel, App with Scenes
- Identifiant généré : `<organisation>-<nom>-<type>` → `telcea-nocstatus-panel`

🕮 La structure d'un plugin de panel

Fichiers clés du squelette

FICHIER	RÔLE
`src/plugin.json`	Identité : type, id, nom, version minimale de Grafana (`>=13.2.0`)
`src/module.ts`	Point d'entrée : panel + options affichées dans l'éditeur
`src/components/SimplePanel.tsx`	Rendu React du panel (données reçues en data frames)
`docker-compose.yaml`	Grafana de développement avec le plugin monté
`tests/`	Tests de bout en bout (Playwright)

- Scripts : `npm run dev` (compilation continue), `npm run build` , `npm run e2e` , `npm run sign`

🕮 Le cycle de développement

Modifier, voir, tester

ÉTAPE	COMMANDE	RÉSULTAT
Dépendances	`npm install`	Modules Node installés
Compilation continue	`npm run dev`	`dist/` régénéré à chaque modification
Grafana de test	`docker compose up`	`http://localhost:3000`, plugin chargé
Tests	`npm run test`, `npm run e2e`	Tests unitaires (Jest), E2E (Playwright)
Signature	`npm run sign` (jeton grafana.com)	`MANIFEST.txt` signé

- Plugin non signé : développement uniquement (`allow_loading_unsigned_plugins`)

Séquence S12.2

Bilan et évaluation

15 min

1. Deux jours en un schéma
2. Objectifs du programme : atteints ?

🕮 Deux jours en un schéma

Du Grafana vide au NOC supervisé

- 6 tableaux de bord, 3 règles d'alerte, 2 équipes, 4 plugins
- Rattrapages et corrections : publiés dans le dépôt `grafana-training`

❓ Objectifs du programme : atteints ?

Auto-positionnement : 1 (non acquis) à 4 (maîtrisé)

OBJECTIF	MODULES	NOTE
Maîtriser les dashboards de Grafana	M04, M05, M06, M08	1 – 4
Savoir configurer un flux de données source	M02, M03, M11	1 – 4
Utiliser les plugins	M07, M12	1 – 4
Alerter et gérer les accès	M09, M10	1 – 4

- QCM final : 20 questions, 15 min (`evaluation-finale.md`)

À retenir

Points clés

- `create-plugin` → React / TypeScript → tests → `build` → signature → publication
- `plugin.json` = identité et compatibilité du plugin
- Les 2 jours : sources → tableaux de bord → variables → plugins → partage → alertes → droits → croisement
- Ressources et documentation officielle : annexes de la formation

QCM d'ancrage (1/3)

Questions 1 à 2

1. Quelle commande génère le squelette d'un plugin ?

- **A.** `npx @grafana/create-plugin@latest`
- **B.** `grafana cli plugins create`
- **C.** `npm init grafana`
- **D.** `docker compose up`

Réponse A — `npx @grafana/create-plugin@latest`.

Révéler la réponse

2. Quel fichier déclare la version minimale de Grafana requise par un plugin ?

- **A.** `README.md`
- **B.** `docker-compose.yaml`
- **C.** `CHANGELOG.md`
- **D.** `src/plugin.json`

Réponse D — `src/plugin.json`, champ `grafanaDependency`.

Révéler la réponse

QCM d'ancrage (2/3)

Questions 3 à 4

3. Avec quelle technologie est écrit le rendu d'un panel ?

- **A.** AngularJS
- **B.** PHP
- **C.** React (TypeScript)
- **D.** Go uniquement

Réponse C — React et TypeScript (Angular retiré en Grafana 12).

Révéler la réponse

4. Quel identifiant suit la convention des plugins ?

- **A.** `NocStatus`
- **B.** `telcea-nocstatus-panel`
- **C.** `panel_noc`
- **D.** `grafana-telcea`

Réponse B — `<organisation>-<nom>-<type>`.

Révéler la réponse

QCM d'ancrage (3/3)

Questions 5 à 6

5. À quoi sert `npm run dev` ?

- **A.** Publier le plugin
- **B.** Signer le plugin
- **C.** Recompiler le plugin à chaque modification
- **D.** Démarrer Prometheus

Réponse C — Compilation continue en développement.

Révéler la réponse

6. Comment charger un plugin non signé dans un Grafana de développement ?

- **A.** Réglage `allow_loading_unsigned_plugins` avec l'identifiant du plugin
- **B.** Impossible
- **C.** Rôle Admin
- **D.** Plugin Angular

Réponse A — `allow_loading_unsigned_plugins` (section `[plugins]`), en développement uniquement.

Révéler la réponse