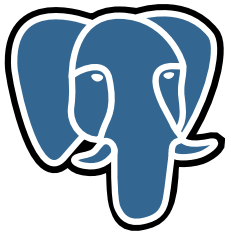
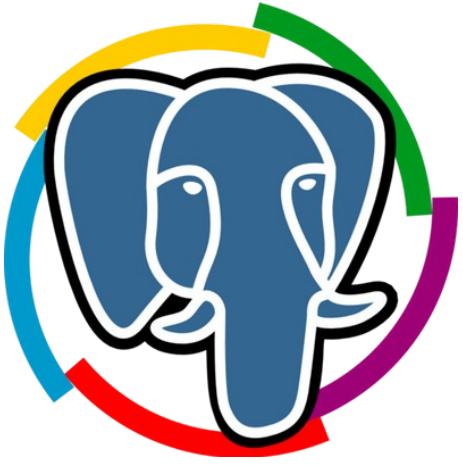


Instrumenter PostgreSQL avec Prometheus



camptocamp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Présentation

- Formation : Scheme, ADA, Java
- Développement PHP, Mixage à la demande, Mastering
- Programmation PL/pgSQL, déplacement de la logique dans PostgreSQL
- Développeur géospatial à Camptocamp
- Infrastructure développeur à Camptocamp
- Open Source
- Formations Docker / Kubernetes / PostgreSQL / PostGIS

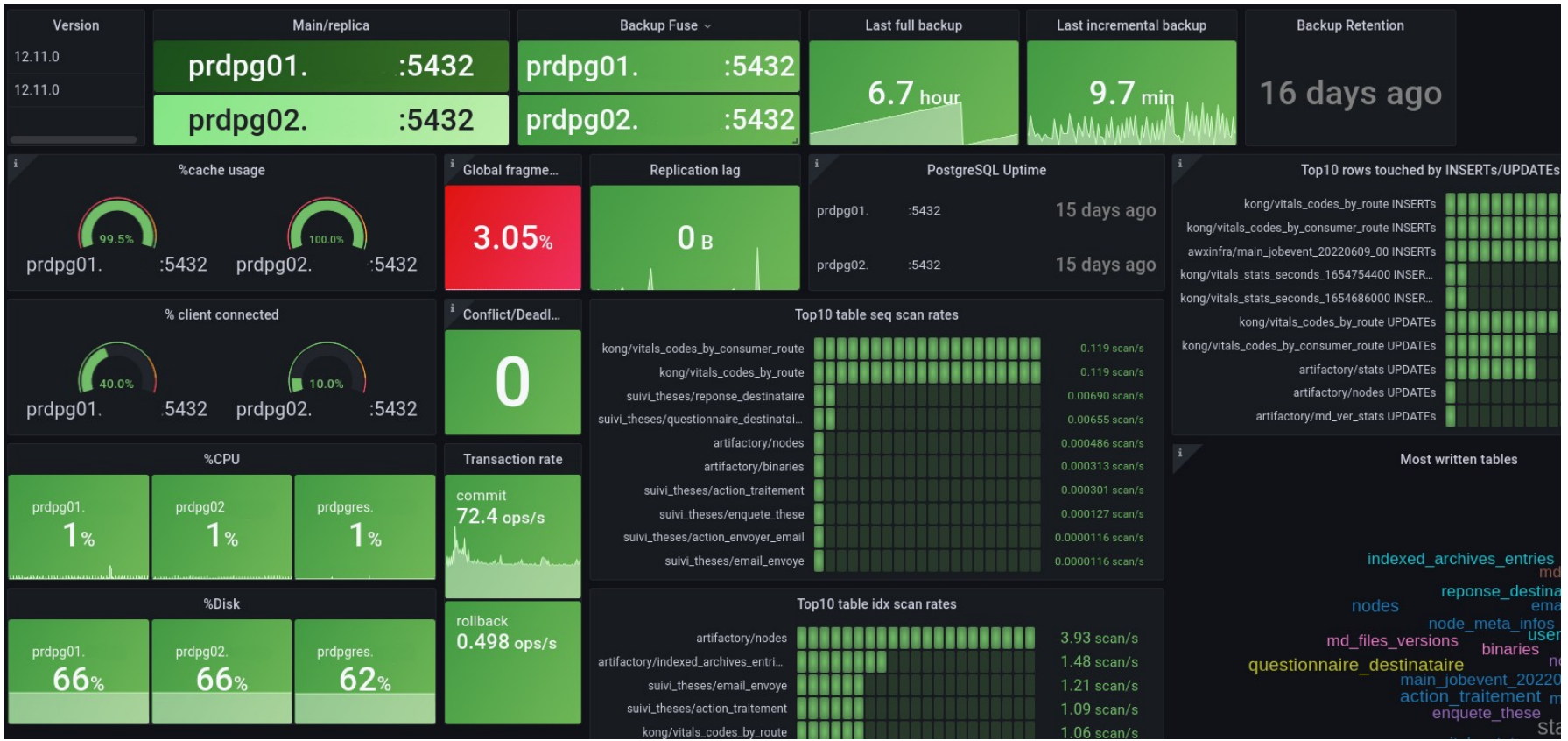
Instrumenter PostgreSQL avec Prometheus

- Observabilité
- Présentation de Prometheus et OpenMetrics
- Métriques du cluster :
  - CPU, RAM, Disques, Réseaux
  - Réplication, Sauvegarde
- Métriques des base de données :
  - tailles, fragmentation,
  - connexions,
  - index,
  - Activité : lecture, écriture

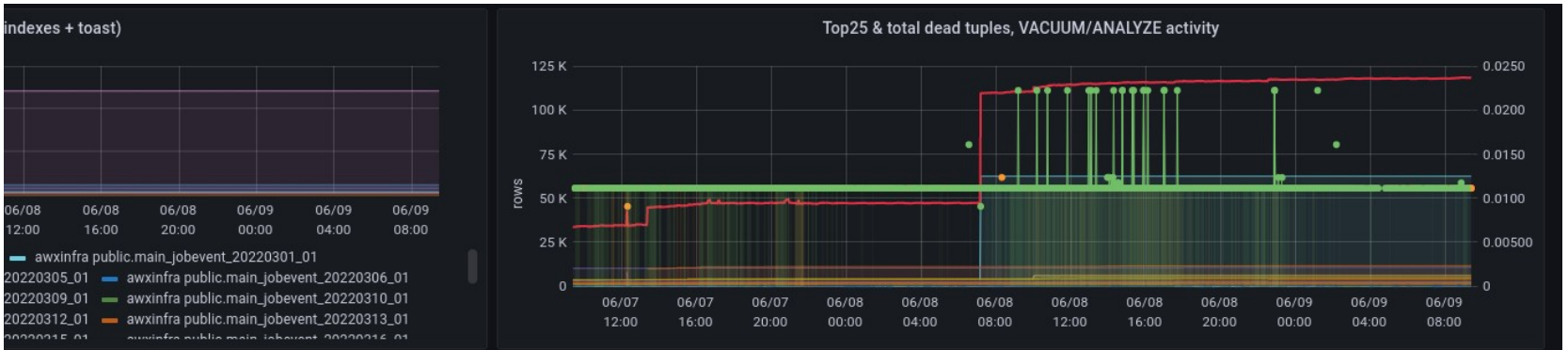
Instrumenter PostgreSQL avec Prometheus

- Métriques des tables :
  - Tailles, données «toastées», nombres d’enregistrements
  - Maintenance, fragmentation
- Métriques des requêtes :
  - requêtes longues, consommatrice de ressources
- Visualisation avec Grafana
- Création d’alerte

Instrumenter PostgreSQL avec Prometheus



Instrumenter PostgreSQL avec Prometheus



Tables Statistics

Live tuples	Dead Tuples ↓	% Fragmentation	Mod since analyse	Last Autovacuum	Last Vacuum	Last Auto Analyse	Last Analyse	Index Size	No
3085657	62371	2.02	26668	-	-	2022-06-08 07:11:01	-	335 MiB	
662	11620	1755	23896	-	-	-	-	96.1 MiB	
58799	10625	18.1	4074	-	-	2022-06-03 12:24:09	-	26.7 MiB	
31644	5943	18.8	1671	2022-06-01 18:01:06	-	2022-06-08 10:00:04	-	10.5 MiB	
33249	4572	13.8	2881	-	-	03:01:31	-	1.26 MiB	
65128	4336	6.66	2509	2022-06-04 02:20:38	-	2022-06-08 12:23:05	-	49.0 MiB	
32325	3687	11.4	400	2022-06-03 22:27:33	-	2022-06-08 22:33:24	-	10.5 MiB	
23250	3402	14.6	1058	-	-	2022-06-07 12:55:09	-	7.90 MiB	
14746	2457	16.7	1465	-	-	2022-06-01 08:02:03	-	7.80 MiB	

Pourquoi Prometheus ?

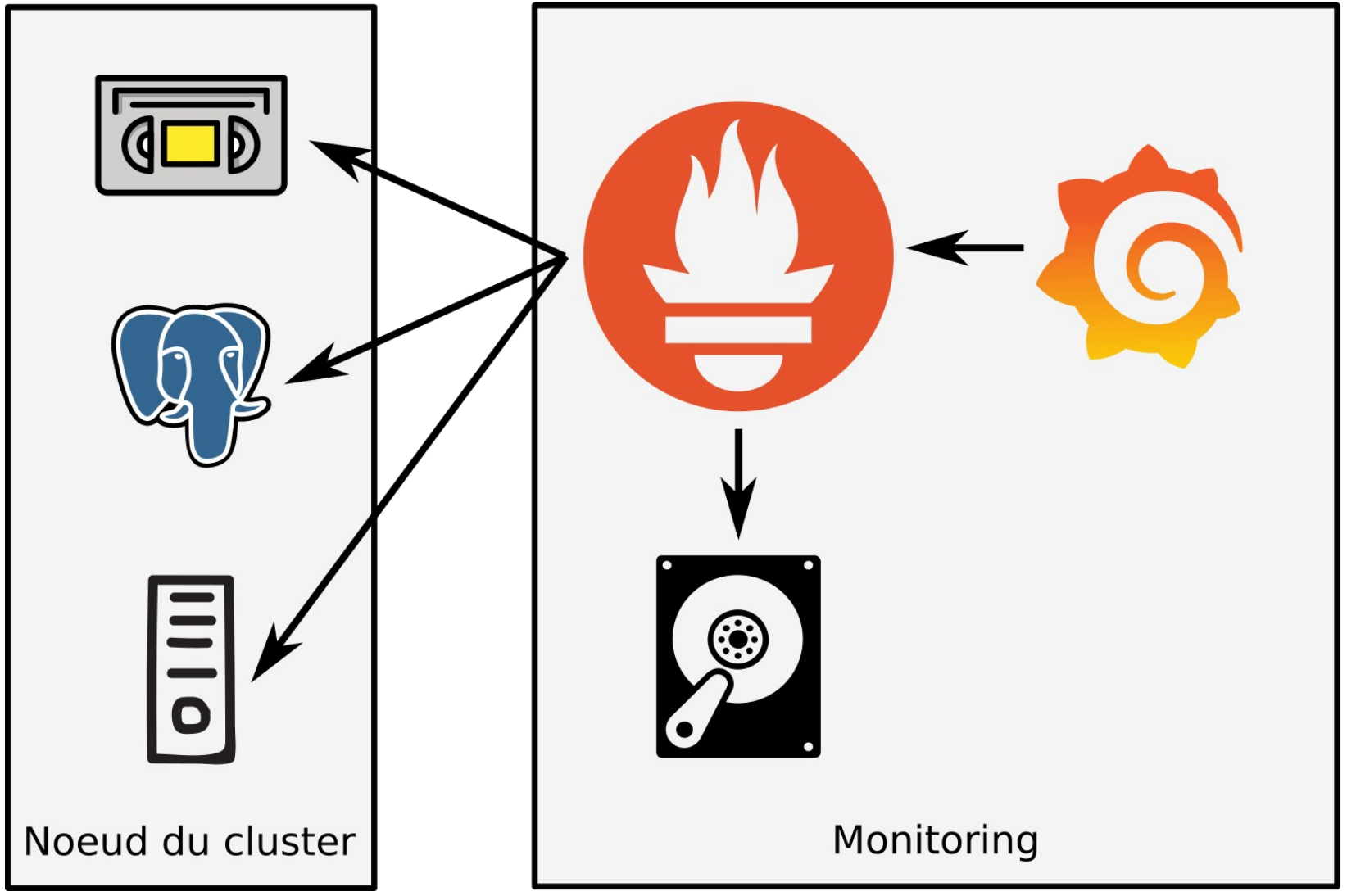
- Outils de monitoring généraliste
- Langage de requête puissant : agrégation, jointure
- Corrélation avec d'autres source données :
  - Métriques système
  - Sauvegarde
  - Cloud Provider
- Utilisable par les développeurs et les administrateurs

Pourquoi Prometheus ?

- Performance
- Pas de stockage en local, centralisation des métriques
- Accès aux indicateurs avec un navigateur
- Métriques personnalisées, facilement extensible
- Analyse après incident
- Pas de valeur instantannées, décalage de 10-30 sec
- Open Source

Fonctionnement de Prometheus

- « Pull » métriques à partir d'« exporter »
- Communication HTTP
- Stocke des séries temporelles
-
- <schema pull target prometheus>


Noeud du cluster
Monitoring

Time Series et Labels

Format OpenMetrics

- Standard en version texte ou binaire
- HTTP GET sur `/metrics`

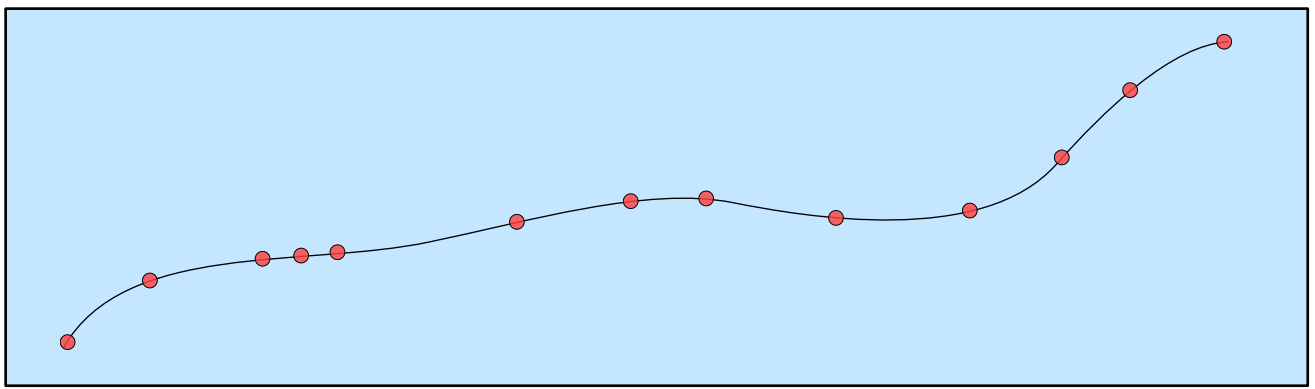
```
# HELP disk_usage_percent Usage of disk in percent (0-100)
# TYPE disk_usage_percent gauge
disk_usage_percent{partition="/"}    63.4
disk_usage_percent{partition="/var"} 37.6
disk_usage_percent{partition="/tmp"} 12.3

# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="GET", code="200"} 1234027 1655884333
http_requests_total{method="POST", code="200"}   1027 1655884333
http_requests_total{method="POST", code="400"}      3 1655884333
```


Compagnons

- Grafana : affichage des métriques (camembert, graphe, VU-mètre)
- Thanos : Stockage au long terme, « downsampling »
- Alertmanager : Routage d’alerte

Traitement à la demande

- Données brutes récupérées régulièrement (< 30sec)
- Dérivée temporelle calculées à la visualisation
- Gestion de la réinitialisation des compteurs
- Gestion de l’irrégularité des intervalles
- Prédiction Linéaire



La concurrence

- pg_view : tourne sur la machine
- pg_activity : statistiques instantanées
- pgcenter
- pganalyse
- pganalyze/collector : exploite la vue pg_stat_plans
- pgSCV : allow custom query
-
- The official exporter win the start war : *1.7k

Extension PostgreSQL

- **pg_stat_statements** : pg_stat_statements tracks all queries that are executed on the server and records average runtime per query "class" among other parameters.
- **pg_stat_plans** : pg_stat_plans extends on pg_stat_statements and records query plans for all executed quries. This is very helpful when you're experiencing performance regressions due to inefficient query plans due to changed parameters or table sizes.
- **pgstattuple** : pgstattuple can generate statistics for tables and indexes, showing how much space in each table & index is consumed by live tuples, deleted tuples as well as how much unused space is available in each relation.
- **pg_buffercache** : pg_buffercache gives you introspection into Postgres' shared buffers, showing how many pages of which relations are currently held in the cache.
- Toutes ces extensions exposent des vues qui peuvent être exploitées par l’exporter PostgreSQL

Métriques standards

- Paramètres de configuration numériques
-

Découverte automatiques des base de données

- Base de connexion
- Autres base de données avec droit de connexion
- Paramètre `master`
- exclude/include

Écrire un exporteur Prometheus

- Librairie Java, Python, …
- On-demand VS Pre-generated
-