# PF-2640
# Thanos Receiver

**Démo Équipe Monitoring du 11/04/2024**

# Le problème

(Yves)

- Prometheus-central explose trop souvent
- Prometheus-central met trop de temps à redémarrer et cela cause l’explosion suivante
- Il faut “vidanger” tous les Prometheus-cluster pour redémarrer
- Prometheus-central est monolithique et supporte mal le passage à l’échelle

# Les solutions

(Yves)

- **Prometheus** avec des *shards*
  - applicable seulement pour le scrapping
  - non applicable pour le remote-receive
- **Mimir** : la cible de NGOT-2025
  - on ne connait pas encore ce logiciel
  - il aurait fallu prévoir un temps d'apprentissage
  - il aurait fallu prévoir une migration de l'existant
- **Thanos-Receive**
  - il faut "juste" déployer un nouveau composant de ce logiciel qu'on connaît déjà
  - il faut "juste" déployer aussi un Thanos-Ruler
  - c'est la solution a priori la plus rapide à mettre en prod
  - c'est la solution nécessitant a priori la plus faible montée en compétence de l'équipe Supervision

# Thanos-Receiver

(Yves)

Thanos-Receiver implémente le protocole Remote-Receive de Prometheus.

Il est possible de distribuer le travail sur plusieurs Thanos-Receivers.

En cas de problème comme avec Prometheus, il suffit donc de “scale up” le nombre de receivers.

# Thanos-Ruler

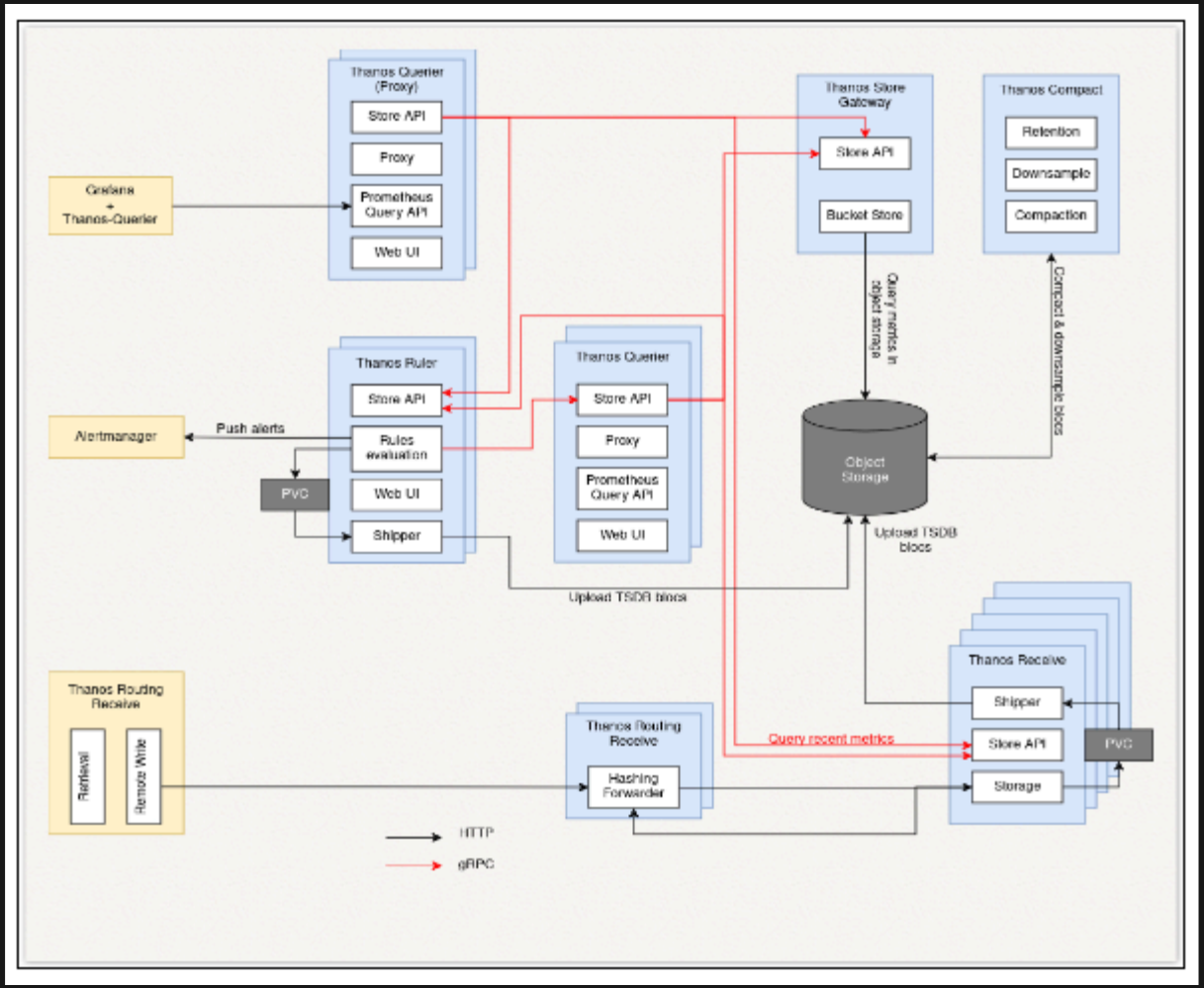
(Yves)

Si Prometheus est déchargé de son rôle de “receiver”, il ne dispose plus de l’ensemble des métriques NGOT. Il n’est donc plus en mesure d’évaluer les rules et d’alerter.

Il faut donc un composant ayant accès à toutes les métriques et pouvant évaluer les rules et alerter : Thanos-Ruler

# Schéma d'architecture (1/2)

(Yves)



# Schéma d’architecture (2/2)

(Nolwenn)

Schéma dans Confluence : noter les blocs

- Receiver et Distributor NEW
- Ruler et Querier dédié NEW
- Querier “proxy” NEW
- Store Gateway : même rôle qu’avant
- Compactor : même rôle qu’avant

# Thanos-Receive “distributor”

(Nolwenn)

Le composant “distributor” n’est rien d’autre qu’un Thanos-Receive. Sa configuration diffère des “receivers” :

- il est stateless (plus rapide) et en HA
- il connait tous les “receivers” : il forwarde aux “receivers” selon un algorithme de hachage “ketama”
- il facilite le “scale up” et minimise le temps d’indisponibilité

# Thanos-Ruler et Thanos-Querier

(Nolwenn)

Thanos-Ruler est déployé par Prometheus-Operator

- bénéficie de la configuration automatique à partir des PrometheusRules
- configuré dans contexts/ngot/kube-prometheus-stack.cue et non thanos.cue.

Thanos-Ruler évalue ses expressions en obtenant les métriques à partir de Thanos-Querier.

- Thanos-Querier dédié (2 replicas pour plus de HA).
- évite les perturbations des consultations via Grafana.

# Thanos-Querier “proxy”

(Nolwenn)

Un composant Thanos-Querier, que nous appelons “proxy” a été ajouté

- parce que le querier de Grafana n’interroge qu’une seule cible (à cause de envoy)
- par effet collatéral, il simplifie l’architecture de façon élégante

Ce composant pourrait également être déployé dans la stack actuelle pour simplifier la configuration du querier (pour pointer sur le sidecar et le store-gateway) mais la valeur ajoutée est moindre.

# Topologie

(Yves)

Sur Prometheus Central :

- un noeud forcé sur une AZ par `topology`
- Prometheus forcé sur ce noeud
- problème : Thanos store, Cloudeye-exporter, Blackbox-exporter, Alertmanager et Kthxbye ne sont pas attachés à une AZ

Sur Thanos-Receiver :

- les pods sont forcés sur des noeuds d'une AZ par `nodeAffinity`
- en place pour Prometheus, Thanos Queriers, Receivers et Distributors, Rulers et Store
- à faire pour Alertmanager et Kthxbye, Blackbox-exporter et Cloudeye-exporter
- affinité hard pour les distributors, soft pour les receivers
- pas besoin de noeuds spécifiques

# Métrologie

(Nolwenn)

Un dashboard spécifique a été créé. Voir également l'étude dans Confluence

Les divers tests montrent globalement que, comparativement à Prometheus Central :

- CPU : la consommation est supérieure
- Mémoire : la consommation est légèrement supérieure
- Disques : la somme des PV/PVC est supérieure

# Tests effectués

(Nolwenn)

Principaux tests effectués :

- scale up, scale down
- redémarrage d’un distributor, d’un receiver
- variations du nombre de distributors, de receivers

Résultats conformes à ce qui est attendu.

Au démarrage, les distributors demandent beaucoup de mémoire (limite mémoire élevée)

Lors d’un incident (scale up/down, redémarrage…) les Prometheus-Cluster “se mettent en incident”.

# Opérations : ce qui est déjà sur la prod

(Yves)

Tous les composants sont déployés :

- cluster `kub-53`
- zone `svc-monitoring-stack-corp-prd-1`
- svc_hint `mon3`

Karma est connecté aux Alertmanagers.

Les Alertmanagers n’envoient aucune notification.

Des silences sont posés parce qu’il n’y a pas encore de MCO sur cette zone.

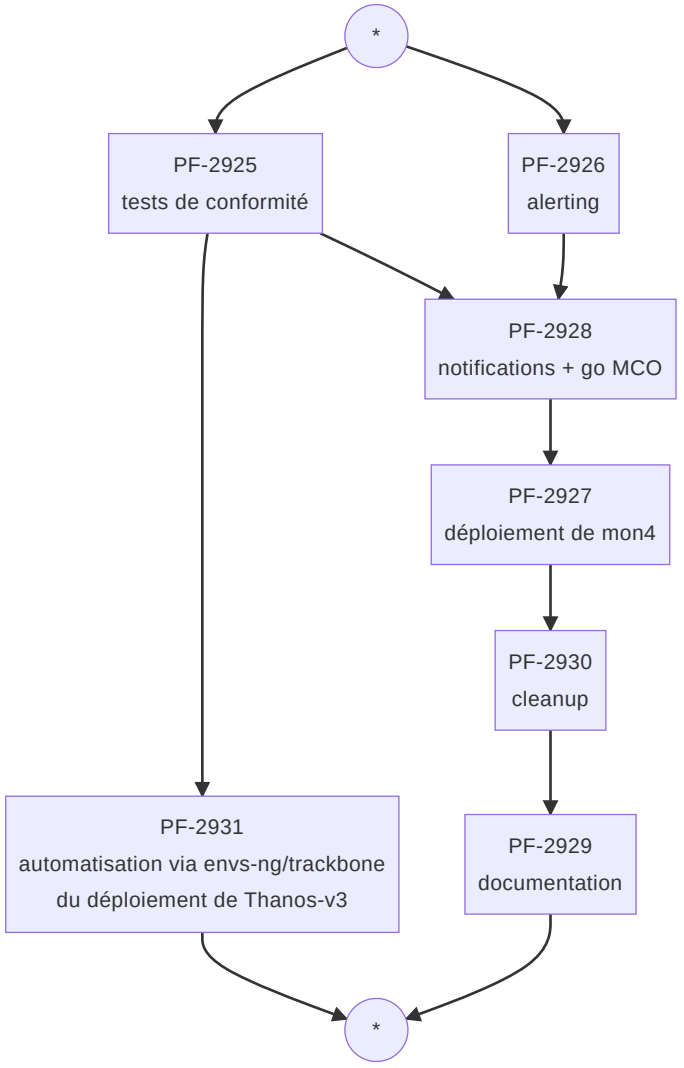
# Opérations : le passage en prod

(Nolwenn)

- Une stack “`mon3`” (zone svc-monitoring-stack-corp-prd-1) a déjà été déployée : CAASCHR-2582
- Il faut terminer les étapes nécessaires au MCO
- Puis passer “`mon3`” en MCO (avec “`mon1`”, “`mon2`”)
- La zone “`mon4`” viendra rapidement après, suivie de la suppression de “`mon1`”, “`mon2`” et d’autres étapes de nettoyage

# Opérations : le passage en prod

(Nolwenn)



# Conclusion

Ceci n’est pas la conclusion ! (Yves)

Après 3 mois de travail, nous avons plein de détails à vous présenter, y compris sur l’existant.

Mais…

- *Prometheus-Central explose trop souvent* : l’origine du problème semble être au niveau de Prometheus-Cluster. Thanos-Receiver n’apporte rien
- Il faut toujours “vidanger” tous les Prometheus-Cluster pour redémarrer (et potentiellement Thanos-Receiver aussi)
- Thanos-Receiver coûte un peu plus cher que Prometheus-Central

Heureusement :

- Thanos-Receiver/Distributors redémarrent plus vite
- Thanos-Receiver/Distributors passent mieux à l’échelle

# Conclusion

(Yves)

La conclusion est à construire ensemble.

Le POC et un MVP sont en place à 90%

Il faut comparer Prometheus Central et Thanos-Receive pour savoir si on valide le POC

Si le POC est validé, une autre présentation aura lieu avec plus de détails techniques.