

PF-2640

Thanos Receiver



2ème démo Équipe Monitoring du 07/05/2024

Contenu de la présentation

(Yves)

- Rappel de la présentation du 11/04/2024
- Rappel de l'architecture
- Principe de déploiement d'une zone
- Méthode pour scale-up/scale-down (démon)
- Difficultés de paramétrage à grande échelle
- Thanos-ruler : le stateful et le stateless. Impact sur Thanos-proxy
- Nombreux petits sujets
- Opérations : image sans shell (démon)
- Conclusion

Rappel de la présentation du 11/04/2024

(Yves)

- exposé du problème amenant à un POC sur Thanos-Receiver
- architecture du POC et description des composants
- description détaillée des Receiver, Distributor, Ruler et Querier “proxy”
- méthode utilisée pour la topologie (et la HA)
- métrologie et dashboard Grafana pour le POC
- principaux tests effectués
- MVP : ce qui est déjà déployé sur la Prod
- reste à faire pour passer en prod (et en MCO)
- argumentaire pour/contre continuer le POC

Rappel de l'architecture

(Yves)

- Schéma d'architecture
- Principaux composants :
 - Écriture : Receiver/Distributor -> Receiver -> S3
 - Lecture : Store Gateway <- Querier *proxy* <- Querier+Grafana
 - Rules : Ruler <- Querier dédié puis selon le cas :
 - -> blocs sur un PV/PVC -> S3
 - -> Alertmanager
 - Compaction/Downsampling/Rétention : Compactor

Principe de déploiement d'une zone (1/3)

(Nolwenn)

composant	Qui déploie ?	envs-ng	Pourquoi ce choix ?
Prometheus	helm/kube-prometheus-stack-v3 + PO	oui	ISO avec l'existant
Thanos-Sidecar	désactivé	oui	on évite une connexion supplémentaire de Grafana + économie de ressources
Thanos-Receive (+distributor)	helm/thanos-v3	non	Pas implémenté dans helm/thanos-v2
Thanos-Ruler	helm/kube-prometheus-stack-v3 + PO	oui	Compatibilité PrometheusRules
Thanos-Store	helm/thanos-v3	non	Ça marche et c'est la cible (MVP)
Thanos-Compactor	helm/thanos-v2	oui	L'utilisation de helm/thanos-v3 nécessite des développements supplémentaires. Ils seront effectués ultérieurement en dehors du POC/MVP
Thanos-Querier (rules)	helm/thanos-v3	non	Ca marche et c'est la cible (MVP)
Thanos-Querier (proxy)	helm/thanos-v3	non	Comme le Thanos-Querier (rules). Il s'agit par contre d'une instance helm distincte.
Thanos-Querier (Grafana)	helm/thanos-v2	oui	Déjà déployé

Principe de déploiement d'une zone (2/3)

(Nolwenn)

- chart umbrella qui a pour dépendance **la chart Bitnami**
 - cette chart est repackagée par nous pour supprimer une dépendance inutile à Minio
 - la méthode de “repackaging” est documentée dans **l'étude** (mais pas dans la doc opérationnelle *docs-internal*)
- déploiement de deux releases de cette chart :
 - thanos-monitoring : pour tous les composants Thanos dont le Querier servant au Ruler
 - thanos-monitoring-querier-proxy : pour le Thanos Querier Proxy servant au Grafana
- commandes de déploiement habituelles : **helm diff upgrade ...** et **helm upgrade --install ...**

Principe de déploiement d'une zone (3/3)

(Nolwenn)

- | Zones | avec Prometheus Central | avec Thanos Receive |
|-----------|---|---|
| nom | svc-monitoring-stack-central-corp-prd-replica-1,
svc-monitoring-stack-central-corp-prd-replica-2 | svc-monitoring-stack-corp-prd-1,
svc-monitoring-stack-corp-prd-2 |
| svc_hint | mon1, mon2 | mon3, mon4 |
| subtype | monitoring-stack | monitoring-stack-corp |
| namespace | monitoring-stack-obs-corp-prd | monitoring-stack-corp-obs-corp-prd |
- nouvelles zones dans [ngot-zones](#)
- liste des configurations pour le nouveau subtype dans [envs.cue](#)
- exemple de configurations modifiées : [alertmanager.cue](#) et [fe_buckets_ngot](#)

Méthode pour scale-up/scale-down

(Nolwenn)

- Opérations :
 - on modifie `replicaCount` dans les `values` de la chart `thanos-v3`
 - `./generate_creds_values.sh ...`
 - `helm diff upgrade ...`
 - `kubectl scale statefulset -n monitoring-stack-corp-obs-corp-stg thanos-receive --replicas=0`
 - `helm upgrade ...`
- Notes :
 - en production, il faut aussi redémarrer tous les Prometheus cluster (sapin de Noël)
 - si scale-down, il faut supprimer le PVC/PV associé à l'ancien réplica

Difficultés de paramétrage à grande échelle

(Yves)

- C'est une équation à plusieurs inconnues :
 - quel flavor de noeud choisir ?
 - combien de replicas de Receivers poser sur un noeud ?
 - combien de Distributors déploie-t-on ?
 - les ingress-controllers supportent-ils bien la charge ? Faut-il en ajouter ?
 - quelles R/L définir ? Pour les Distributors ? Pour les Receivers ?
 - quelle volumétrie PV/PVC choisir ?
 - quand on atteint la limite, sont-ce les R/L qui sont trop petites ou faut-ils d'autres réplicas ?
- Avec des éléments perturbateurs :
 - un simple scale-up est long et pose problème aux Prometheus sources
 - un scale-down à 0 suivi d'un redéploiement est curieusement plus efficace
 - quasi chaque modification (sur un pod Distributor ou Receiver) crée un incident "sapin de Noël"
 - le rythme de croisière n'est atteint qu'au bout de 2 jours environ. La plupart des tests ont dû être observés pendant 48h avant d'en tirer des conclusions
 - à petite échelle (staging) les résultats sont un peu différents d'à grande échelle (prod)

Thanos-ruler : le stateful et le stateless

Impact sur Thanos-proxy

(NoIwenn)

Ce qu'il faut savoir :

- on déploie en Stateful (parce qu'on ne pouvait pas le faire en Stateless lors du démarrage de l'étude) ;
- le Stateless est plus fiable (pas de PV/PVC) ;
- simplifie la configuration des Queriers *proxy* ;
- permet la HA ;
- en pratique, on n'a pas de recording rules, le PV/PVC semble inutile.

Que faire ?

- actuellement, cela fonctionne ;
- ce serait mieux en Stateless.

Est-ce un détail ou devrions-nous envisager de redéployer en Stateless ?

Nombreux petits sujets

(Nolwenn)

- l'option `partial_response_strategy` dans les PrometheusRules
- l'option `--tsdb.too-far-in-future.time-window` dans Thanos Receive pour refuser les métriques dans le futur
- nos choix de labels de déduplication et le semi-problème d'override des labels sur les prometheus centraux
- le replication factor pour dupliquer les métriques
- le **Thanos Receiver Controller**
- l'algorithme "ketama" vs "hashmod" : celui par défaut est fortement déconseillé
- le multi-tenant avec Thanos
- l'UI dans Thanos-Store et les nombreux "streams"
- l'UI dans Thanos-Query et les nombreux composants auxquels il est connecté

Opérations : image sans shell

(Yves)

- problème : les images Thanos bitnami n'ont pas de shell, cela peut-être un frein lors du debug ou lors d'une modification à faire sur un disque
- solution : utiliser les Ephemeral Containers
- documentation : [KB/Kubernetes/Déboguer avec les Ephemeral Containers](#)
- exemple avec un pod receive :

```
IMAGE=alpine:latest
NAMESPACE=monitoring-stack-corp-obs-corp-stg
POD=thanos-receive-0
TARGETCONTAINER=receive

kubectl -n "${NAMESPACE}" debug \
    -it \
    --image="${IMAGE}" \
    --profile=general \
    --target="${TARGETCONTAINER}" \
    "${POD}" -- sh
```

- note : il existe une autre méthode basée sur un plugin de `kubectl` que nous n'avons pas documentée.

Conclusion

(Yves)

- Plus de 3 mois sur PF-2640 Thanos-Receiver
- Nous avons subi...
 - Certains problèmes viennent de la source Prometheus (erreurs 409, out_of_order_samples...)
 - Il a fallu upgrader Prometheus-Operator/Prometheus pour... `sample_age_limit` qui n'a pas tenu ses promesses
 - Thanos Receiver/Ruler/Distributor est une solution très récente : il faut des charts Helm, Prometheus-Operator et Thanos très à jour.
 - Thanos vs Prometheus : Prometheus n'a pas été conçu pour fonctionner à grande échelle mais il s'en sort très bien... Que faire ?
 - Le paramétrage de Thanos à grande échelle est long et très difficile.
- Nous avons appris...
 - Thanos est un bel outil qui va nous rassurer face à la montée en charge
 - L'architecture de Thanos est bien conçue et celle de Mimir va lui ressembler (sauf le Compactor)
 - Le dashboard conçu pour le POC a énormément servi à voir même ce qu'on n'avait pas prévu de voir.
 - Le protocole Remote-Write évolue (une version 2.0 en préparation...)
 - L'univers Bitnami a ses charmes (images sans shell, charts imbriquées à repackager...). Nous savons connecter un Ephemeral Container à un pod existant.
 - Si on va un jour au Pérou, envisager une visite de la "montagne Arc-En-Ciel" !
- Merci à tous !
- La suite : [PF-2932](#) et sa suite de PF !



