

PF-2942

Upgrade Prometheus et Prometheus-Operator



Démo Équipe Monitoring du 29/04/2024

Contenu de la présentation

(Yves)

- Les ChangeLogs
 - Prometheus
 - Prometheus-Operator
 - la chart Helm
- Démo : ScrapeConfig
- Démo : ScapeClass
- Métrologie
- Prometheus et Thanos-Receiver
- Conclusion

ChangeLog : Prometheus

(Yves)

ChangeLog 2.42.0 -> 2.51.2

- modifications récurrentes sur plusieurs versions
 - “performance improvements”
 - plusieurs améliorations relatives à remoteWrite (source et destination), dont `sample_age_limit` ou des modifications de codes d'erreur HTTP (503/422 -> 499 dans certains cas)
 - refactor puis bugfixes sur les labels (à partir de 2.43.0)
- **2.48.0 [ENHANCEMENT]** Scraping: Add configuration option for tracking staleness of scraped timestamps. [#13060](#)
- **2.49.0 [BUGFIX]** Mixins: Fix url filter on remote write dashboards. [#10721](#)
- **2.50.0 [FEATURE]** Add automatic memory limit handling. Activated using the feature flag. `auto-gomemlimit` [#13395](#) : envisager un PF d'étude ?

ChangeLog : Prometheus-Operator

(Notwendig)

ChangeLog 0.63.0 -> 0.73.1

- **0.64.0** : nouveau CRD PrometheusAgent
- **0.65.0** : nouveau CRD ScrapeConfig
- **0.66.0** : option `keep_firing_for` dans les PrometheusRules
- **0.71.0** : ajout des métriques
`prometheus_operator_status_update_errors_total` et
`prometheus_operator_status_update_operations_total`
 - Cela permet de savoir quand il y a des échecs sur la modification du status des ressources
 - Nous avons une alerte déployée (`PrometheusOperatorStatusUpdateErrors`) utilisant ces métriques, elle va maintenant fonctionner.
- **0.71.0** : ajout d'un paramètre `maximumStartupDurationSeconds` dans le CRD Prometheus
- **0.72.0** : ajout d'un objet `ScrapeClass` dans le CRD Prometheus
- **0.73.0** : ajout du champ `sampleAgeLimit` dans le CRD Prometheus

ChangeLog : chart Helm

(Nolwenn)

ChangeLog 45.7.1 -> 58.1.3

- le ChangeLog est long mais reporte globalement les modifications de Prometheus-Operator.
- la [section Upgrading...](#) du [README.md](#) mentionne :
 - 47.x to 48.x : les CRDs sont déplacés dans une subchart dédiée
 - 51.x to 52.x : quelques changements pour Thanos
- nos changements notables sont :
 - possibilité de configurer Thanos Ruler pour envoyer les recording rules en RemoteWrite sur les Thanos Receive (mode stateless)
 - ajout du scraping des métriques du config reloader de Prometheus, nous avons une alerte `ConfigReloaderSidecarErrors` déployée utilisant ces métriques, elle va maintenant fonctionner.

ScrapeConfig simple

(Yves)

```
apiVersion: monitoring.coreos.com/v1alpha1
kind: ScrapeConfig
metadata:
  name: pf-2942-simple
  namespace: monitoring
  labels:
    cloudservicesfactory/managed-by: corp-cluster
spec:
  staticConfigs:
    - labels:
        job: pf-2942-simple
      targets:
        - demo.do.prometheus.io:9100
```

Grafana : node_exporter_build_info{goversion!="go1.19.3"}

ScrapeConfig basicAuth (1/2)

(Yves)

Générer le secret (on va fédérer test01 qui expose son API) :

```
#!/bin/bash

export export VAULT_ADDR=https://vault.infra-stg.caascad.com; vault token s=$(vault read /secret/zones/fe/svc-monitoring-stack-client-test01/prometheus)
cat <<EOF > secret.yaml
apiVersion: v1
kind: Secret
metadata:
  name: pf-2942-basic-auth
  namespace: monitoring
data:
  password: "$(echo -n "${s}" | jq -r .data.password | tr -d '\n' | base64)"
  user: "$(echo -n "${s}" | jq -r .data.username | tr -d '\n' | base64)"
type: Opaque
```

ScrapeConfig basicAuth (2/2)

(Yves)

```
apiVersion: monitoring.coreos.com/v1alpha1
kind: ScrapeConfig
metadata:
  name: pf-2942-authconfig
  namespace: monitoring
  labels:
    cloudservicesfactory/managed-by: corp-cluster
spec:
  staticConfigs:
    - labels:
        job: pf-2942-authconfig
      targets:
        - prometheus.obs-test01.csfpriv.com
  basicAuth:
```

Grafana : `stackdriver_exporter_build_info{ticket="pf-2942"}`

ScrapeClass (1/2)

(Noiwenn)

⚠ Attention ⚠ : fonctionnalité “expérimentale” (mais cool)

Sur `kub-10003` (par exemple) :

```
kubectl -n monitoring edit prometheus cluster-  
prometheus
```

```
spec:  
  scrapeClasses:  
    - name: prez20240425  
      relabelings:  
        - action: replace  
          replacement: pf-2942  
          targetLabel: testprez
```

ScrapeClass (2/2)

(Nolwenn)

Sur `kub-10003` (par exemple) :

```
kubectl -n logging edit smon promtail
```

```
spec:  
  scrapeClass: prez20240425
```

Grafana : `promtail_build_info{cluster="kub-10003"}`

Métrologie

(Yves)

	CPU	Mémoire (en Mo)
prometheus-operator	1,5m -> 3,5m (pics à 150m)	69-133 -> 34-54
config-reloader	0,6m -> 1m	18-26 -> 30-45
prometheus-cluster	230m -> 140m	1300-2800 -> 1050-2350
prometheus-central	8 -> 6,5	64Go -> 56 Go

Grafana CPU - Grafana Mémoire

Prometheus et Thanos-Receiver

(NoIwenn)

- Paramètre `sample_age_limit`
 - l'upgrade a provoqué des “incidents”. Ce paramètre n'a pas eu l'effet escompté et provoque même un bug. Il a été temporairement supprimé.
 -  Mimir, Thanos-Receiver et Prometheus fonctionnent de la même façon : si des métriques sont “bloquées” dans l'agent (Prometheus-cluster sur NGOT), le problème existera aussi sur les nouvelles stacks.
- Il est maintenant possible de choisir le mode de déploiement de Thanos-Ruler (stateful vs stateless)
- Les versions de la chart kube-prometheus-stack, Prometheus-Operator et Thanos-Ruler sont en phase.

Conclusion

(NoIwenn)

- L'upgrade était nécessaire pour avancer sur le sujet Thanos-Receiver (PF-2640).
- Les avancées espérées (paramètre `sample_age_limit`, bugfix divers) sont décevantes.
- Notre satisfaction principale est d'avoir Prometheus-Operator et Prometheus à jour (ou presque).

