Agents Natifs GCP

Objectifs

- Tester l’agent natif de GCP pour collecter logs et métriques
- Essayer et analyser les fonctionnalités du tableau de bord natif de GCP
- Modifier la configuration pour les logs et métriques

GCP cloud monitoring

Cloud monitoring offre une solution pour collecter, analyser et exploiter les données...

Les agents proposés sont :

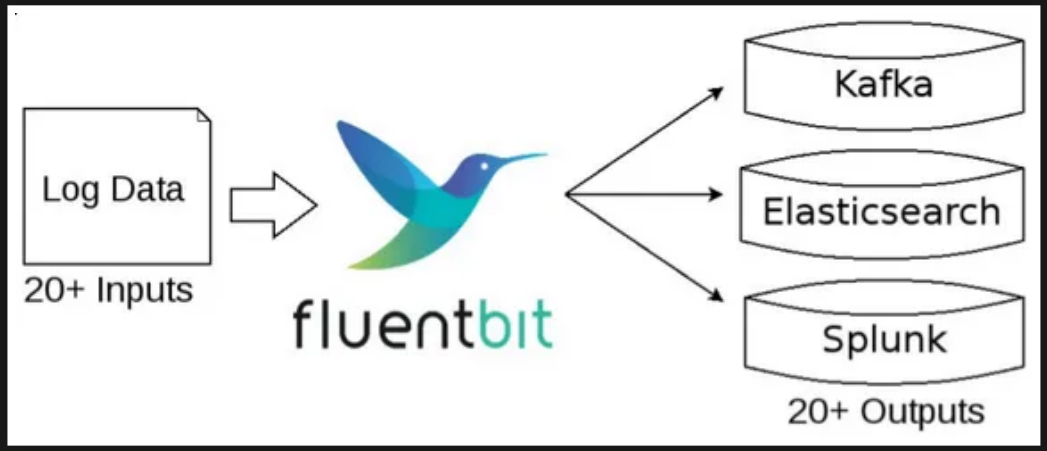
- Agent Ops (par défaut)
- Legacy agent (obsolète)

En combinant la journalisation et les métriques dans un seul agent, l’agent Ops utilise **Fluent Bit** pour les journaux, qui est compatible avec la journalisation à haut débit, et le collecteur **OpenTelemetry** pour les métriques.

Fluentbit

Fluentbit est un projet open source créé par l'équipe TreasureData et sous la responsabilité de CNCF (Cloud Native Computing Fundation). Il permet la gestion des logs dans les VMs et conteneurs. Etant basé sur fluentd, fluentbit apporte plusieurs avantages:

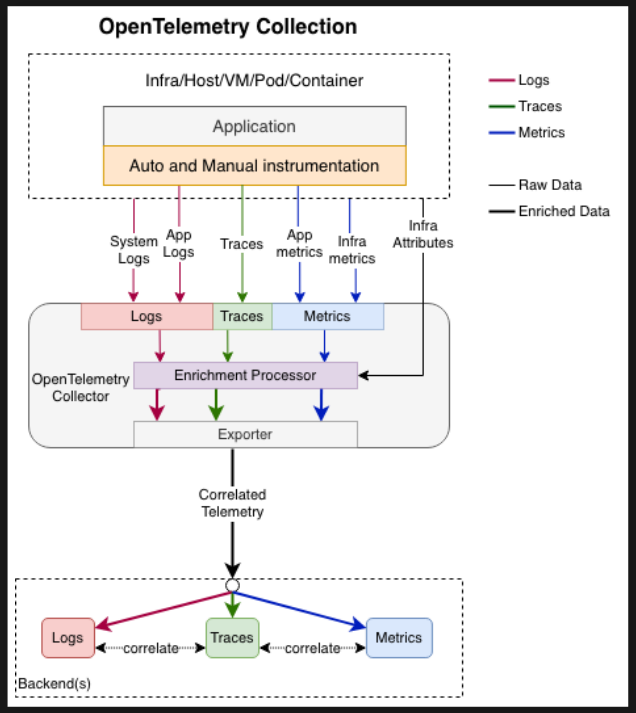
- consommation mémoire moindre
- Pas de dépendances externes
- journalisation haut débit


Log Data
20+ Inputs
fluentbit
Kafka
Elasticsearch
Splunk
20+ Outputs

OpenTelemetry

[OpenTelemetry](https://opentelemetry.io/) est un projet de l'incubateur CNCF.

Il est utilisé pour orchestrer, collecter, générer et exporter des données (métriques, logs et traces) afin d’analyser les performances et les comportements des logiciels.


OpenTelemetry Collection
Infra/Host/VM/Pod/Container
Application
Auto and Manual instrumentation
Logs
Traces
Metrics
Raw Data
Enriched Data
System Logs
App Logs
Traces
App metrics
Infra metrics
Infra Attributes
Logs
Traces
Metrics
OpenTelemetry Collector
Enrichment Processor
Exporter
Correlated Telemetry
Logs
correlate
Traces
correlate
Metrics
Backend(s)

Configuration de l’agent OPS

L’agent Ops utilise une configuration par défaut intégrée et immuable. On peut cependant y ajouter notre fichier de configuration après redémarrage de l’agent. Les composants de la configuration sont :
Receivers : cet élément décrit ce qui est collecté par l’agent.
Processors : cet élément décrit comment l’agent peut modifier les informations collectées.
Service : cet élément associe les récepteurs et les processeurs pour créer des flux de données, appelés pipelines.

La configuration de l’agent Ops est à modifier dans les fichiers suivant :
Pour Linux : /etc/google-cloud-ops-agent/config.yaml
Pour Windows : C:\Program Files\Google\Cloud Operations\Ops Agent\config\config.yaml

```yaml
logging:
  receivers:
    syslog:
      type: files
      include_paths:
      - /var/log/messages
      - /var/log/syslog
  service:
    pipelines:
      default_pipeline:
        receivers: [syslog]
metrics:
  receivers:
    hostmetrics:
      type: hostmetrics
      collection_interval: 60s
  processors:
    metrics_filter:
      type: exclude_metrics
      metrics_pattern: []
  service:
    pipelines:
      default_pipeline:
        receivers: [hostmetrics]
        processors: [metrics_filter]
```

Configuration de métriques

L'élément receivers contient un ensemble de définitions de récepteur. Un récepteur désigne où extraire les métriques, par exemple cpu et memory. Un récepteur peut être partagé entre plusieurs pipelines.

Chaque récepteur doit avoir un identifiant, RECEIVER_ID, et inclure un élément type. Les types valides sont les suivants :

- hostmetrics
- iis (Windows uniquement)
- mssql (Windows uniquement)

```
metrics:
  receivers:
    hostmetrics:
      type: hostmetrics
      collection_interval: 10s
```

Configuration de la journalisation

L’élément receivers contient un ensemble de récepteurs, chacun étant identifié par un RECEIVER_ID. Un récepteur décrit comment récupérer les journaux (par exemple, en affichant les dernières lignes des fichiers, via un port TCP, ou depuis le journal des événements Windows).

Les types valides sont les suivants :
files : collecter des journaux en affichant les dernières lignes des fichiers sur le disque.
fluent_forward : collecter les journaux envoyés à l’aide du protocole Fluent Forward via TCP.
syslog : collecter syslog via TCP ou UDP.
tcp : collecter les journaux au format JSON en écoutant un port TCP.
windows_event_log (Windows uniquement) : collecter les journaux d’événements Windows.
systemd_journald : collecter des journaux à partir du service systemd-journald.

```
receivers:
  RECEIVER_ID:
    type: files
    ...
  RECEIVER_ID_2:
    type: syslog
    ...
```

Cas pratique : surveillance de l’applications Nginx

Prérequis

- Nginx installé avec Stub-status activé
- configuration de l’agent OPS (opentelemetry)

Nginx étant l’une des applications pré configurées pour l’agent ops, un minimum de configuration suffit pour exporter à la fois les métriques et les logs.

La configuration de l’agent ops pour nginx

```
sudo tee /etc/google-cloud-ops-agent/config.yaml > /dev/null << EOF

# CONFIGURATION DES LOGS - ACCES ET ERREURS
logging:
  receivers:
    nginx_access:
      type: nginx_access
    nginx_error:
      type: nginx_error
  service:
    pipelines:
      nginx:
        receivers:
          - nginx_access
          - nginx_error
# CONFIGURATIOND DES METRIQUES
metrics:
  receivers:
    nginx:
      type: nginx
      stub_status_url: http://127.0.0.1:80/nginx_status
  service:
    pipelines:
      nginx:
        receivers:
          - nginx
# REDEMARRAGE DE L'AGENT
sudo service google-cloud-ops-agent restart
EOF
```

Filtrage des métriques

- ajout d'un processors
- un seul filtre possible (exclude_metrics)

```
processors:
  metrics_filter:
    type: exclude_metrics
    metrics_pattern:
    - agent.googleapis.com/nginx.connections_current/*
service:
  pipelines:
    nginx:
      receivers:
        - nginx
      processors:
        - metrics_filter
```

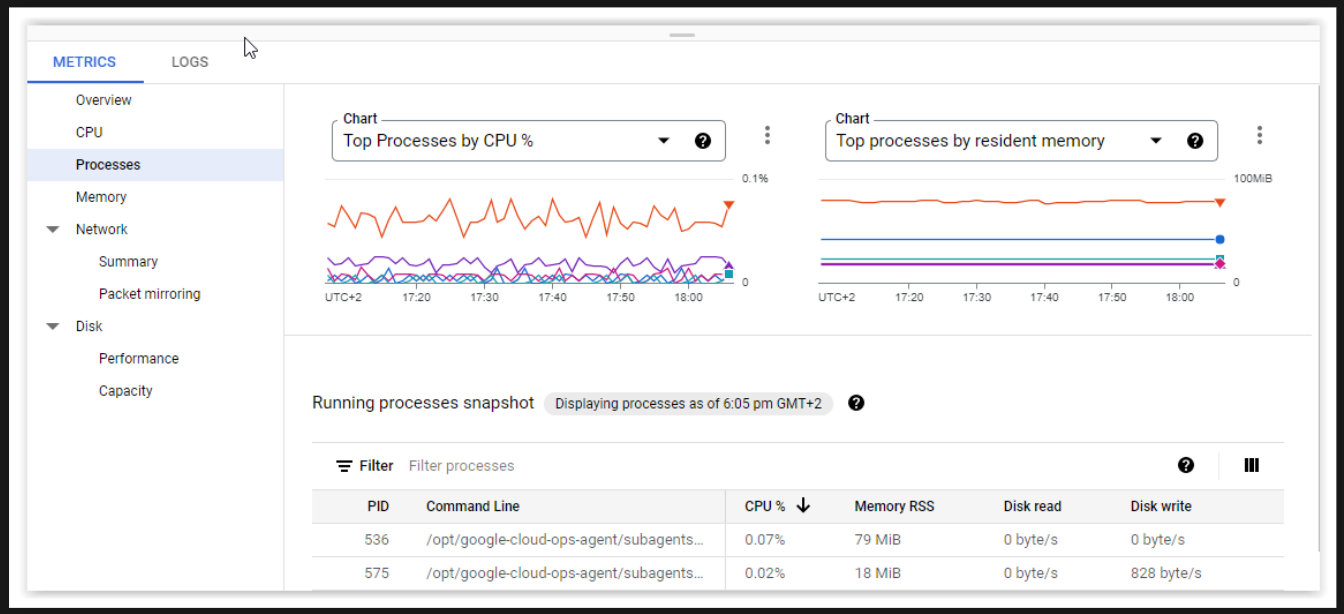
Filtrage de la journalisation

- ajout d'un processors
- plusieurs filtres possibles (parse_json, parse_regex, exclude_logs)

```
processors:
  filtrage:
    type: exclude_logs
    match_any:
    - 'jsonPayload.message =~ "user cloudwatt"'
service:
  pipelines:
    auth_pipeline:
      receivers:
      - auth
      processors:
      - filtrage
```

Empreinte sur la machine virtuelle

GCP fournit d’origine un tableau de bord permettant d’observer les processus de la machine.



Limitations

- Pas de support multiline pour les stacktrace mais un MR est ouvert dans ce sens
- Impossible de créer des métriques custom en utilisant directement le collector opentelemetry
- Un nombre limité d’applications préconfigurées (voir la liste https://cloud.google.com/monitoring/agent/ops-agent/third-party)
- Pas de lograte pour les logs de l’agent Ops

NB: GCP travaille déja sur toutes ces limitations

Conclusion

- facilité d’installation et de configuration
- des tableaux de bord pré configurés et nativement accessibles
- configuration uniforme entre windows et linux
- empreinte mémoire minime
- documentation complète
- repo github très actif

Liens

documentation confluence
documentation upstream de l’agent
repo github