

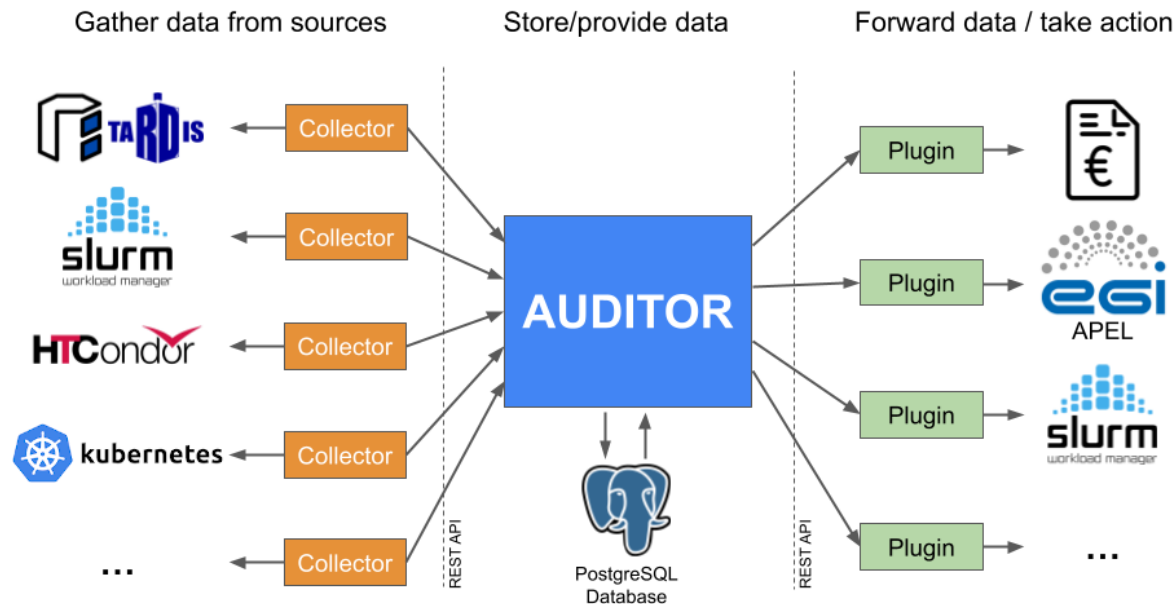


Auditor

Retour d'expérience au LLR

Présentation

- **A**ccounting **D**ata Handling **T**oolbox For **O**ppportunistic **R**esources
- Agrégateur ou source de données d'accounting
- Prometheus



Installation

- Compilation (Cargo), rpm, Docker
- 2 parties : Auditor (@LLR : *Docker Compose / Portainer*) & collecteur (@LLR : *rpm htcondor-collector*)
- rpm : GitHub (<https://github.com/ALU-Schumacher/AUDITOR/releases>)



Installation – Auditor/Docker

- 2 temps
 - préparation DB
 - utilisation
- Portainer
 - création stack complète (Auditor et healthcheck commentés)
 - « migrate »
- Volumes :
 - fichier de configuration (yaml)
 - certificats

Installation – Auditor/Docker

- API :
 - /health_check : vide, code retour 200 si OK
 - /metrics : utilisé par Prometheus
 - /records ou /record/<record_id> : détails des/du record

```
# HELP num_records_database Number of records in the Auditor database
# TYPE num_records_database gauge
num_records_database 38
# HELP num_records_database_per_group Number of records in the Auditor database
# TYPE num_records_database_per_group gauge
num_records_database_per_group{group_id="alice"} 1
num_records_database_per_group{group_id="atlas"} 26
num_records_database_per_group{group_id="cms"} 1
num_records_database_per_group{group_id="lhcb"} 8
# HELP num_records_database_per_site Number of records in the Auditor database
# TYPE num_records_database_per_site gauge
num_records_database_per_site{site="GRIF-IJCLAB"} 23
num_records_database_per_site{site="GRIF-LLR"} 15
# HELP num_records_database_per_user Number of records in the Auditor database
# TYPE num_records_database_per_user gauge
num_records_database_per_user{user_id="alis"} 1
num_records_database_per_user{user_id="atlp000"} 23
num_records_database_per_user{user_id="atls"} 3
num_records_database_per_user{user_id="clasjls"} 2
num_records_database_per_user{user_id="cms"} 1
num_records_database_per_user{user_id="lhcp000"} 8
# HELP target_info Target metadata
# TYPE target_info gauge
target_info{service_name="unknown_service",telemetry_sdk_language="rust",telemetry_sdk_name="opentelemetry",telemetry_sdk_version="0.23.0"} 1
```

Installation – htcondor-collector

- rpm
 - /etc/auditor/auditor_htcondor_collector.yml (configuration)
 - /opt/auditor_htcondor_collector/ (venv + certificats => attention au owner)
 - user: auditor-htcondor-collector / group: auditor
 - service auditor_htcondor_collector

HEP score

- ClassAd HTCondor

- récupération des données CPU
- worker node
 - détecte son CPU, fait la correspondance avec le score HS23
 - enregistré en tant que ClassAd « Machine » (type CRON, « One-Shot »)
- condor-ce
 - récupère la donnée HS23 depuis la machine pour l'inscrire dans le Job
 - Auditor lit le job, et peut parser sur la valeur HS23
 - exemple : MachineAttrHEP_score0 = "HEPscore23:478.733"

```
LastSuspensionTime = 0
LeaveJobInQueue = false
MachineAttrApelScaling0 = undefined
MachineAttrApelSpecs0 = undefined
MachineAttrCpus0 = 8
MachineAttrHEP_score0 = "HEPscore23:478.733"
MachineAttrSlotWeight0 = 8
MaxHosts = 1
MemoryProvisioned = 20480
```

Problèmes rencontrés

- mise en place Docker
 - certificats
 - RBAC
-
- côté positif : les devs sont à l'écoute, plutôt réactif, bon échange

Merci pour votre attention

BACKUP

Docker compose

```
services:
  auditor:
    image: aluschumacher/auditor:0.10
    depends_on:
      auditor_db:
        condition: "service_healthy"
    ports:
      - "8000:8000"
      - "8443:8443"
    environment:
      - AUDITOR_DATABASE__DATABASE_NAME=auditor
      - AUDITOR_DATABASE__HOST=host.docker.internal
      - AUDITOR_DATABASE__PASSWORD=dbpassword
      - AUDITOR_DATABASE__PORT=5432
      - AUDITOR_DATABASE__REQUIRE_SSL=false
      - AUDITOR_DATABASE__USERNAME=postgres
      - AUDITOR_ENVIRONMENT=production
      - AUDITOR_RBAC_CONFIG__ENFORCE_RBAC=false
    volumes:
      - /opt/auditor/config.yaml:/auditor/config.yaml
      - /opt/auditor/certs:/auditor/certs
    restart: "unless-stopped"
```

```
entrypoint:
  - /auditor/entrypoint.sh
  - auditor
  - /auditor/config.yaml
extra_hosts:
  - "host.docker.internal:host-gateway"

auditor_db:
  image: postgres:18
  healthcheck:
    test: "pg_isready -U postgres -d auditor"
    interval: 30s
    timeout: 60s
    retries: 5
    start_period: 80s
  ports:
    - "5432:5432"
  environment:
    - POSTGRES_USER=postgres
    - POSTGRES_PASSWORD=dbpassword
    - POSTGRES_DB=auditor
  volumes:
    - /opt/auditor/docker_data/auditor_db:/var/lib/postgresql
  restart: "unless-stopped"
```