

# Monitoring/Reporting

---

## CloudWatch

### CloudWatch Dashboards

- Tableau de bord entièrement customisable.
- Possibilité de partage (y compris sur une autre région) sans donner accès à son compte.
- Jusqu'à 3 dashboard avec 50 metrics gratuitement. Au delà 3\$/mois par dashboard supplémentaire.

### CloudWatch Metrics and Anomaly Detection

#### CloudWatch Metrics

- Par défaut les mesures sont faites toutes les 5 mn.
- Pour un faible coût il est possible de le faire toutes les mn.
- Possibilité de créer ses propres mesures (au sein d'une région).

#### Anomaly Detection

- Détection automatique d'une situation anormale à l'aide des metrics pour générer une alarme.
- Fonctionne par apprentissage (learning machine) en analysant l'historique des metrics.

### CloudWatch Alarms

- Action automatique déclenchée sur dépassement de seuil de métrique.
- 3 niveaux : OK, ALARM (dépassement du seuil), INSUFFICIENT\_DATA (métrique non disponible ou insuffisante)

### CloudWatch EventBridge

- Pour connecter à différents services pour faire du monitoring temps réel

#### Rules

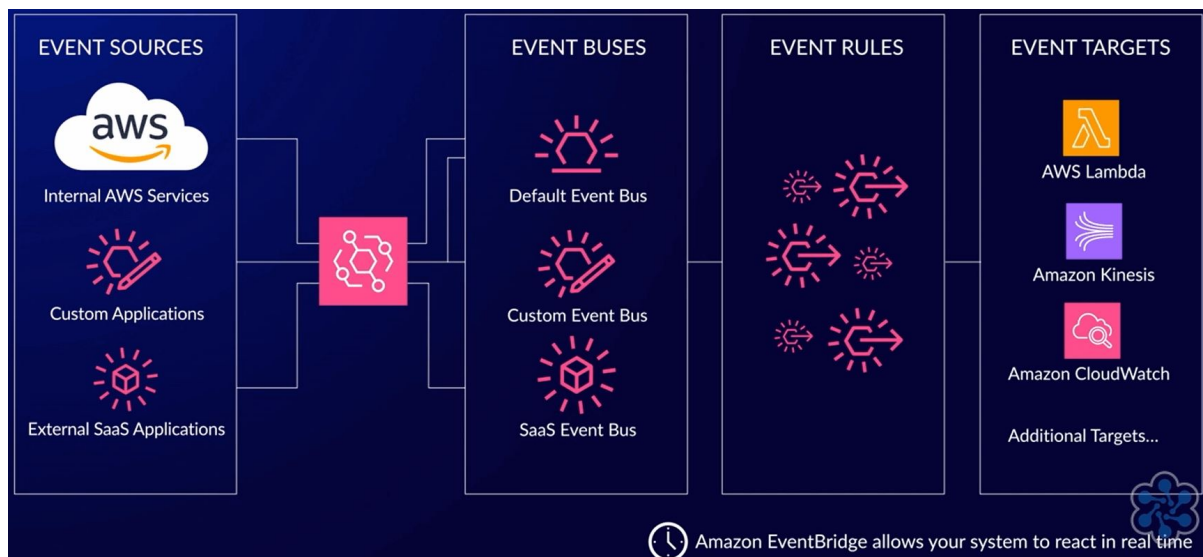
- Filtre pour diriger les events vers les bonnes targets (dans la même région).

#### Targets

- Destinataires des events (AWS Lambda, SQS, Kinesis, SNS, ...)
- Les events sont au format JSON

## Event buses

- Composant qui reçoit les events de l'application
- Les rules sont associés à un event bus (jusqu'à 100 rules par event bus).
- Il existe un event bus par défaut pour recevoir les events des services AWS



## CloudWatch Logs

- Supervision centralisée des logs des différents services AWS

## Unified CloudWatch Agent

- Agent qui peut être installé sur différents OS (CentOS, RedHat, Ubuntu, Debian, Windows Amazon Linux)

## CloudWatch Insights

### Log Insights

- Analyse des logs

### Container Insights

### Lambda Insights

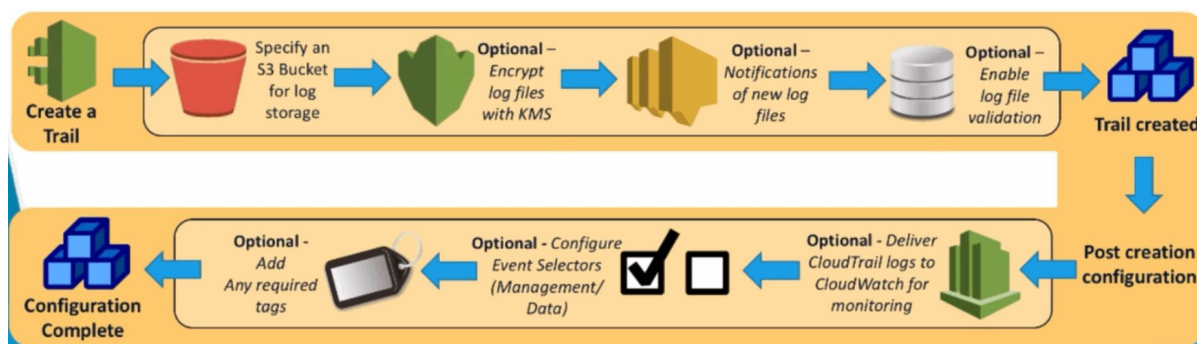
[Haut de page](#)

## Cloudtrail

- Enregistre toutes les requêtes vers l'API (au sein du compte AWS)
- Chaque requête est vu comme un event
- Envoyé vers un S3 bucket ou vers CloudWatch Logs

### Composants

- Trails : configuration correspondant aux requêtes que l'on souhaite capturer
- S3 : stockage par défaut des logs (envoyé 15mn après)
- Logs : fichiers créés toutes les 5 mn avec les requêtes définis dans le trail
- KMS : encryption optionnelle lors du stockage dans le S3 bucket
- SNS : notification optionnelle pour notifier quand de nouveaux fichiers de logs sont stockés
- CloudWatch Logs : redirection optionnelle en plus du S3 pour monitoring
- Event Selectors : permet d'ajouter un niveau de sélection du type de requête à capturer
- Tags : permet d'ajouter ses propres metadata au trail
- Events : chaque requête est enregistrée comme un event
- API Activity Filters : filtres de recherche pour créer, modifier ou effacer des requêtes dans l'historique d'activité.

[Haut de page](#)

## AWS Config

Fonctionne au sein d'une région

- Capturer les modifications des ressources
- Enregistrer un historique des configurations
- Prendre un snapshot des configurations
- Inventorier les ressources
- Envoyer des notifications sur des modifications

- Fournir des informations sur qui a fait des modifications et quand
- Analyse de sécurité
- Identifier les relations entre les ressources

## Composants

### AWS ressources

Objets qui peuvent être créés, modifiés, ou effacés

### Configuration Item (CI)

Fichier JSON. Créé dès qu'un changement intervient sur une ressource.  
Contient 5 types d'informations :

- Metadata : informations concernant le CI (version, MD5hash, date, ...)
- Attributes : informations concernant la ressource
- Relationships : description des relations avec d'autres ressources
- Current Configuration : liste les informations de la ressource
- Related Events : event ID de CloudTrail qui est à l'origine de ce CI

### Configuration Stream

Les CI sont envoyés à un configuration stream qui est de la forme d'un SNS topic. Egalement utilisé dans les cas suivants :

- delivery d'un fichier configuration history
- démarrage d'un configuration snapshot
- modification de l'état de "compliance" d'une ressource
- démarrage d'une évaluation
- échec de notification par AWS config

### Configuration History

Produit un historique des modifications en utilisant les CI. L'historique peut également être envoyé toutes les 6h dans un S3 bucket.

### Configuration snapshot

Prend un snapshot de toutes les ressources configurées dans une région, un CI de chaque ressource peut être stocké dans un S3 bucket.

## **Configuration recorder**

Moteur du service qui enregistre tout les changements et génère les CIs. Activé par défaut lors de la configuration de AWS config. Peut être suspendu.

## **Config rules**

Chaque règle est une Lambda fonction qui va évaluer la modification de la ressource par rapport à la règle et envoyer éventuellement un message à la configuration stream via SNS.  
Il y a déjà des règles prédéfinies utilisables.

## **Ressource relationship**

Identification des relations entre les ressources.

## **SNS Topic**

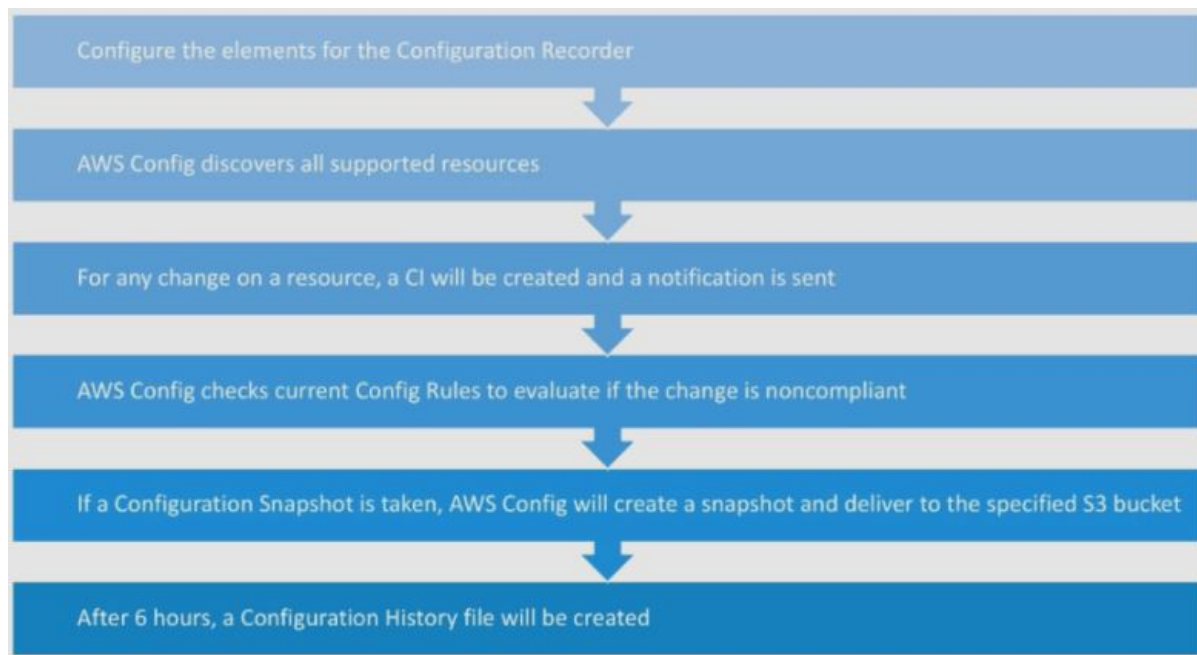
Utilisé comme configuration stream pour envoyer des notifications.

## **S3 bucket**

Pour stocker les fichiers Configuration history toutes les 6h ainsi que tout les snapshots.

## **AWS Config Permissions**

Il faut associé un IAM role pour autoriser les accès.



---

[Haut de page](#)

## Amazon Athena

Permet de faire des requêtes SQL pour chercher/filtrer des données dans les logs stocké dans un S3 bucket.

---

[Haut de page](#)

## Bills and Cost

### Billing dashboard

### Cost explorer

### Cost and Usage Reports

### Budgets

---

[Haut de page](#)

From:

<https://wiki.iot-acs.fr/> - Wiki

Permanent link:

<https://wiki.iot-acs.fr/doku.php?id=all:bibles:aws:presentation:7-monitoring>

Last update: **2024/06/14 11:10**

