NoSQL, systèmes distribués et passage en production de projets Data

Thierry GAMEIRO MARTINS

Séances

1. Introduction et prise en main d'Onyxia

2. **Le stockage des données en NoSQL**

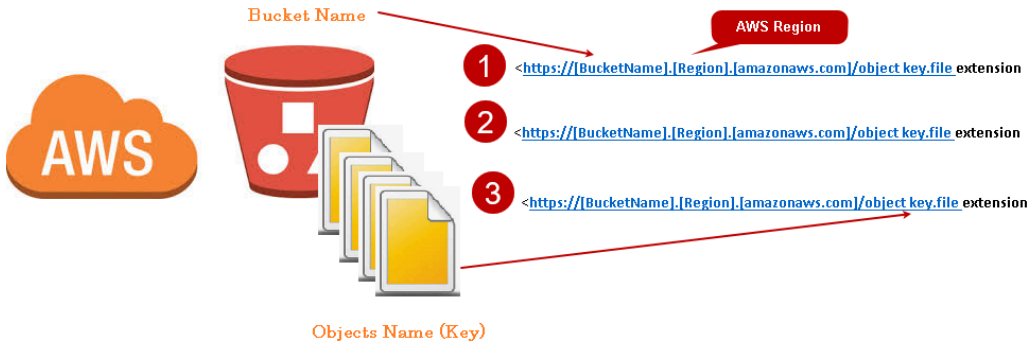
3. Les systèmes de traitement distribués
4. Le passage en production
5. Orchestration et pratique DevOps
6. Déploiement conteneurisé sous Kubernetes

Amazon S3 et les formats de données

amazon
S3

S3 pour *Simple Storage Service* est un Service de stockage répliqué et évolutif sous forme d'objets sur le web

- Stockage (réplication, cycle de vie, opérations par lots, etc.)
- Diversité (tout types de données, site web, etc.)
- Accessibilité (protocole HTTP)
- Gestion des accès (via les ACL)



Bucket

- Conteneur pouvant contenir des fichiers
- Gestion des accès et des actions possibles
- Peut servir des sites web statiques
- Peut fournir des évènements selon des actions

Objets

- Les objets sont des fichiers accessibles par une URL sur le web
- Les objets peuvent êtres chiffrés sur le disque
- Manipulation des fichiers comme dans un système de fichier classique (`mv`, `ls`, `rm`, `cp`, etc.)
- Toutes les manipulations se font par le protocole `HTTP`

Utilisation avec S3cmd

`S3cmd` est un client écrit en python pour manipuler le stockage S3

Pour créer ou supprimer un bucket

```
s3cmd mb s3://BUCKET
s3cmd rb s3://BUCKET
```

Pour lister les buckets ou les fichiers d'un bucket

```
s3cmd ls [s3://BUCKET[/PREFIX]]
```

> S3cmd se base sur la librairie python `boto3`

Envoyer un fichier

```
s3cmd put FILE [FILE...] s3://BUCKET[/PREFIX]
```

Télécharger ou supprimer un fichier

```
s3cmd get s3://BUCKET/OBJECT LOCAL_FILE
s3cmd rm s3://BUCKET/OBJECT
```

Copier un fichier d'un bucket à un autre

```
s3cmd cp s3://BUCKET1/OBJECT1 s3://BUCKET2[/OBJECT2]
```

Expiration des objets

Une lifecycle configuration est composé d'un ensemble de règle (format JSON ou XML) avec comme propriétés :

- **id** : identifiant de la règle
- **status** : activé ou désactivé
- **filter** : permet de filtrer des objets avec des conditions (par exemple par un prefix, par tag, taille d'objet, etc.)
- **lifecycle action** : le type d'action à effectuer (transition, expiration, annuler les upload en plusieurs partie incomplets, etc.)

```
<LifecycleConfiguration>
    <Rule>
        ...
    </Rule>
    <Rule>
        ...
    </Rule>
</LifecycleConfiguration>
```

```
<LifecycleConfiguration>
  <Rule>
    <ID>Transition and Expiration Rule</ID>
    <Filter>
       <Prefix>tax/</Prefix>
    </Filter>
    <Status>Enabled</Status>
    <Transition>
      <Days>365</Days>
      <StorageClass>S3 Glacier Flexible Retrieval</StorageClass>
    </Transition>
    <Expiration>
      <Days>3650</Days>
    </Expiration>
  </Rule>
</LifecycleConfiguration>
```

Gestion des permissions sur les objets

- **sid**: nom de l'ACL
- **resource** : L'Amazon ressource name concernée
  - `"Resource": "arn:aws:s3:::bucket_name"`
  - `"Resource": "arn:aws:s3:::bucket_name/*`
- **actions** : action effectuée (`s3:ListBucket` par exemple pour lister les objets)
- **effect** : l'effet recherché, `allow` ou `deny` , par défaut `deny`
- **principal** : le sujet de la policy

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:user/Dave"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::awsexamplebucket1/*",
                "arn:aws:s3:::awsexamplebucket1"
            ]
        }
    ]
}
```

Utilisation avec S3cmd

Pour appliquer une bucket policy depuis un fichier local ou la supprimer

```
s3cmd setpolicy FILE s3://BUCKET
s3cmd delpolicy s3://BUCKET
```

Pour afficher les informations associés à un bucket (policy, etc.)

```
s3cmd info s3://BUCKET[/OBJECT]
```

Envoyer un fichier

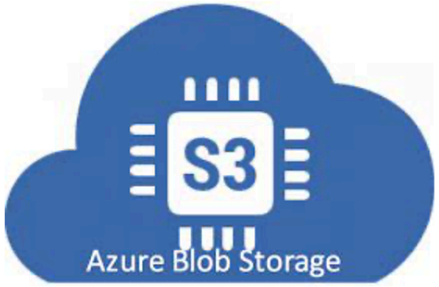
```
s3cmd put FILE [FILE...] s3://BUCKET[/PREFIX]
```

Appliquer une lifecycle policy depuis un fichier local, ou la visualiser / supprimer

```
s3cmd setlifecycle FILE s3://BUCKET
s3cmd getlifecycle s3://BUCKET
s3cmd dellifecycle s3://BUCKET
```

Les implémentations de s3 sont nombreuses :

- Pas toutes les fonctionnalités toujours présentes
- Il existe des versions open-source



Formats de données

Le choix d’un format de données pour du stockage doit se faire selon les critères suivants :

- le public
- la finalité (traitement, analyse, diffusion)
- la volumétrie
- la compatibilité des outils

Limites des formats usuels

Les formats usuels (CSV, JSON, XML) sont utiles pour :

- Le traitement de faibles volumes de données
- La diffusion de données

Limités pour le traitement de données volumineuses

- **Non-compressés** : espace disque élevé
- **Orientés ligne** : peu adaptés aux traitements analytiques

Le format Parquet

Les propriétés

- **Orienté colonne**
  - Adapté aux traitements analytiques
  - Conçu pour être écrit une fois mais lu fréquemment
- **Optimisé**
  - Forte compression
  - Rapidité de lecture du fichier
  - Gestion native des méta-données

Le partionnement

- Division en blocs des données selon un critère
- Optimise la lecture pour certaines requêtes