# Atelier Accès aux Données 2024
# Datalake CNES / Protocole S3

10-11/10/2024 – Plouzané, Ifremer

Cyril GERMINEAUD (CNES)



contact@odatis-ocean.fr | www.odatis-ocean.fr

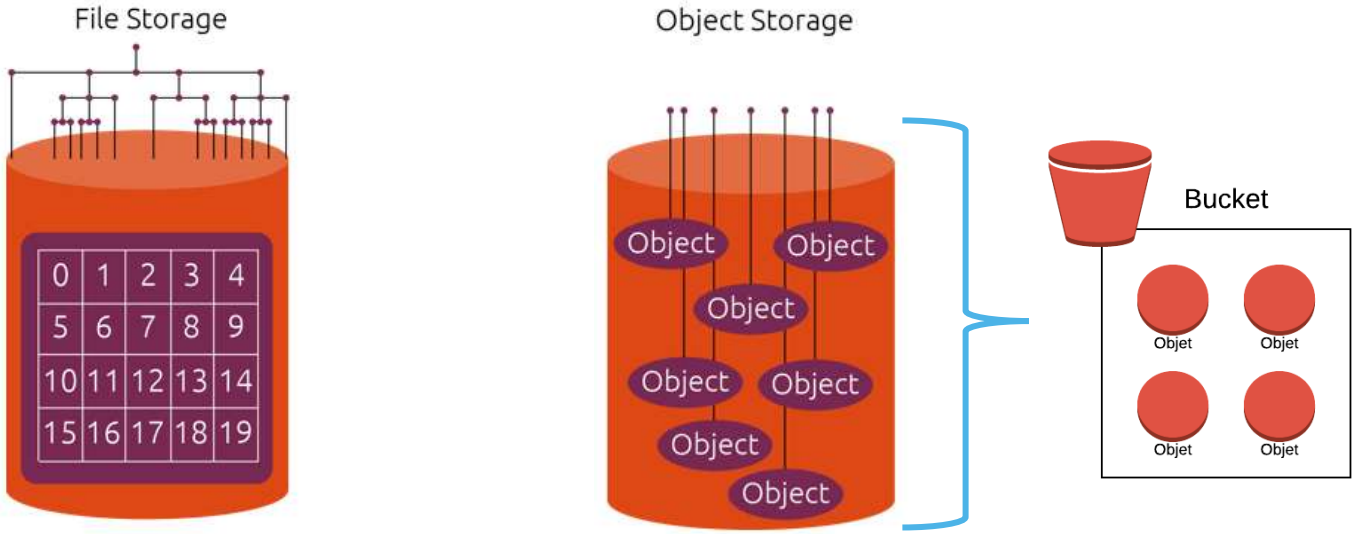
# Le Datalake du CNES : Protocole S3

**Utiliser le Datalake => utiliser le protocole S3**

système de fichiers : **POSIX**

**VS**

AWS S3 (*Simple Storage Service*) : stockage **objet**

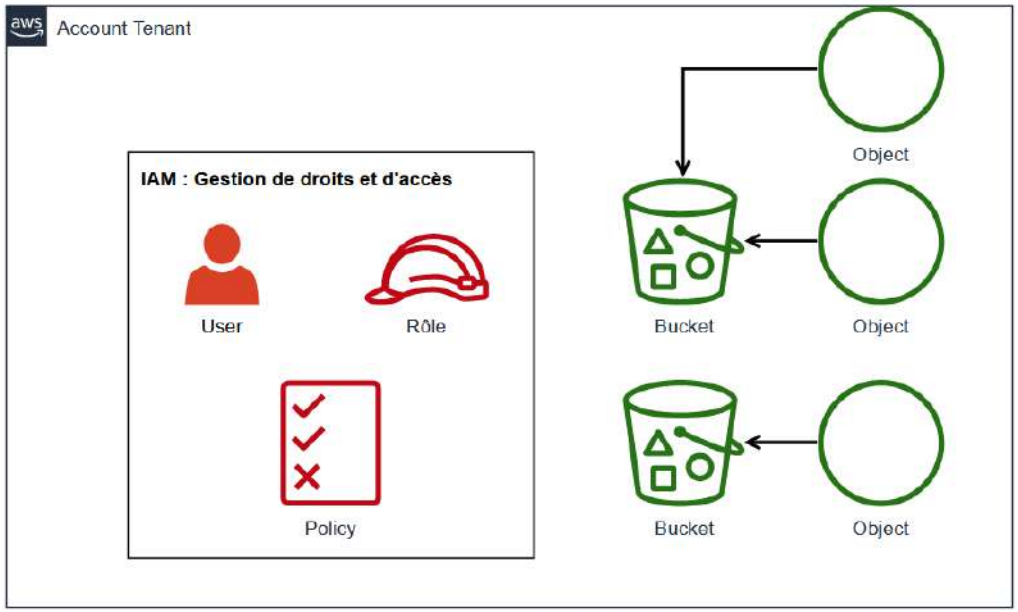


# Le Datalake du CNES : Protocole S3

## Concepts Importants

- Organisation des données :
  - Tenant / Projet (Account au sens AWS)
  - Buckets à l'intérieur des tenants
  - Des objets placés à plat au sein des buckets

- Les accès via entités IAM :
  - Policy : les politiques d'accès
  - User/Group : utilisateurs/groupes IAM authentifiés avec une paire de clé
  - Rôle : Entité détenant des droits d'accès à des ressources (bucket/objet) et pouvant être endossé temporairement



# Le Datalake du CNES : Protocole S3

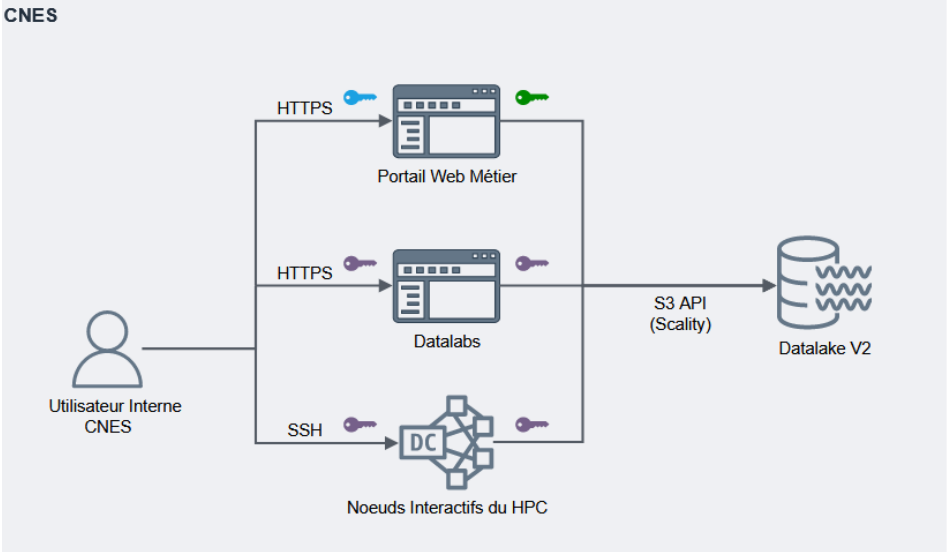
## Credentials S3 == paire de clés

- aws_access_key_id
- aws_secret_access_key
- *aws_session_token* (si utilisation d'un rôle)

## Différents modes d'authentification

- Via credentials permanents :
  - IAM User (pas lié au compte IPA)
  - À destination des projets / producteurs de données

- Via credentials temporaires (max 12h) :
  - En endossant un rôle grâce à son compte IPA
  - À destination de tout les utilisateurs




Légende :

- Utilisateurs gérés par l'application métier. Inconnu du Datalake V2.
- Utilisateurs gérés par l'annuaire IPA CNES. A accès aux services du SIS.
  Utilisateurs disposants des groupes IPA permettant un accès au Datalake V2.
- Utilisateurs techniques disposants des groupes permettant un accès au Datalake V2.

# Les services AWS

**Uniquement 3 services AWS sont disponibles sur le Datalake (gestion stockage S3)**


**Gestion de ressources**
[ bucket et object ]


**Gestion de entités IAM**
[ user, group, role et policy ]
*Chaque account possède son propre IAM*


**Gestion de credentials temporaires avec une durée de 12h**

**Le Datalake respecte les APIs standards AWS**
la documentation AWS S3 est donc une excellente base pour commencer

# Le Datalake du CNES : Protocole S3

## Accès au Datalake

- Un endpoint : https://s3.datalake.cnes.fr
- Une région : us-east-1

## S'authentifier au Datalake

- Paire de clés *aws_access_key_id et aws_secret_access_key*
- Token *aws_session_token (si on utilise un rôle)*

## Configuration des credentials

- Variables d'environnement *AWS_ACCESS_KEY_ID AWS_SECRET_ACCESS_KEY AWS_SESSION_TOKEN*
- Profils *en configurant les fichiers ~/.aws/credentials et ~/.aws/config*
- Directement dans le client (NON recommandé)

**Les credentials permettent d'accéder et d'avoir la visibilité aux ressources d'un seul Account.**
**Les accès sont limités aux autorisations fournis via les politiques (IAM policy).**
**Par défaut, aucune autorisation (implicit deny).**

# Le Datalake du CNES : Usage du S3

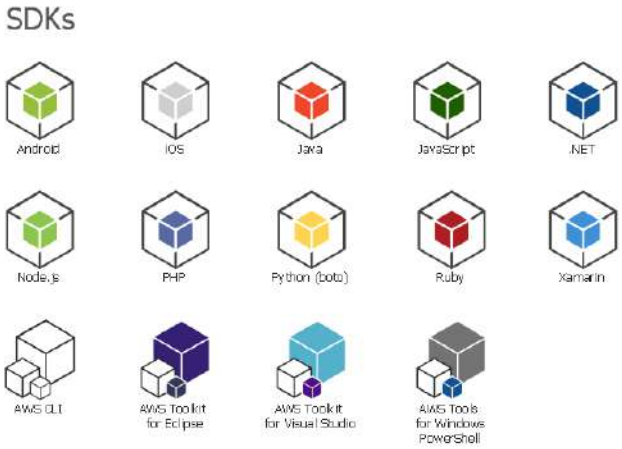
## Quel est le périmètre d'un utilisateur ?

| | Dans son périmètre | Hors périmètre (équipe Datalake) |
|---|---|---|
| AWS IAM | • Connaitre les principes de la gestion des droits<br>• Pouvoir demander si son besoin le nécessite des clés permanentes (Read-only, Read/write) | Création de IAM User, Rôle ou policy |
| STS<br>AWS STS (alternate) | • Connaitre les principes<br>• Utiliser les scripts présents dans /softs/tools/datalake/ ou librairie assumerole | Paramétrer les rôles<br>• en gestion d'accès (Policy)<br>• en gestion d'autorisation pour les comptes IPA |
| Amazon S3 | • Choisir son client S3<br>• Paramétrer ses authentifications (paire de clé)<br>• Manipuler les objets<br>• **Tier 3** : Restaurer les objets et superviser le tier de l'objet | Création des buckets<br>**Tier 3** : Paramétrer les règles ILM |

# Clients S3 et particularités du Datalake

**Toujours utiliser un client S3 reconnu ou librairies de traitements**

- **Utiliser un client S3 qui permet de manipuler les données**
  - Des APIs haut-niveaux :
    - Simuler les commandes POSIX (ls, cp …)
    - Uploads/downloads partitionnés
  - Des APIs bas niveaux : get-object, put-object, head-object, list-object…
- **Prendre en compte les caractéristiques du stockage objet**
  - très performant en lecture mais comparativement peu performant en écriture.
  - organisation des données : metadonnées personnalisable, tags …
  - formats de données plus ou moins adaptés (en terme de performance) pour le S3




DASK

pandas

GDAL

**Les opérations autorisées (GetObject, PutObject …) dépendent des autorisations acquises par l'entité IAM (credentials) utilisée**