CLYSO

Powered by ceph

Ceph S3 - Placement Dynamique et Rétention Optimisée

Storage Classes, LUA et LifeCycle Policies

Frédéric Nass | Senior Ceph Engineer @CLYSO

Content

01. **Classes de stockage**
Placement targets & pools

02. **Placement dynamique**
LUA scripting

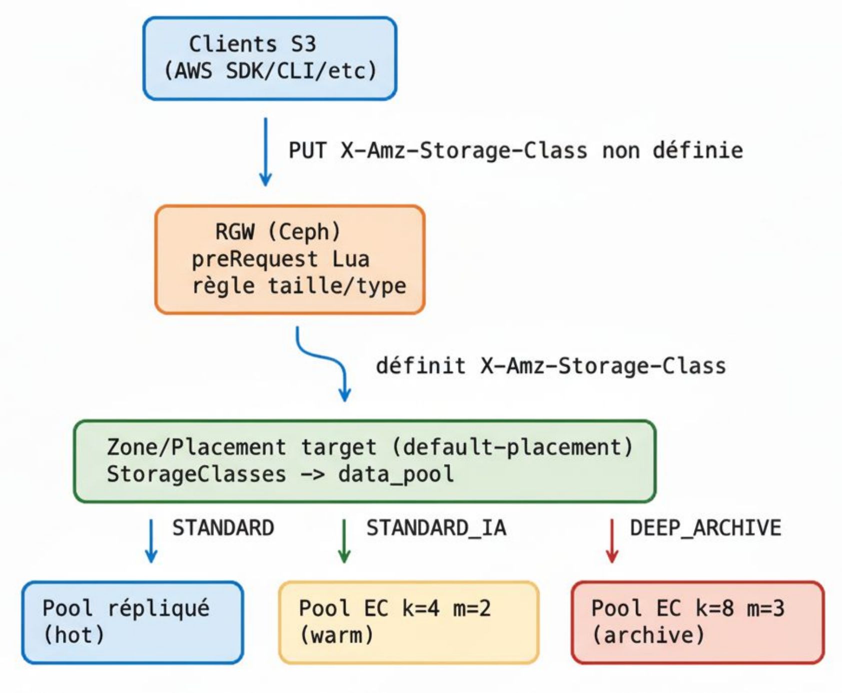
03. **Rétention optimisée**
LifeCycle Policies

04. **Démo**
Écriture d'objets S3

Classes de stockage

- Une ou plusieurs dans un stockage S3
- Associées à un modèle de placement de donnée : replication, erasure coding, compression
- Optimisent le placement des objets S3 en fonction de différents critères : performance, fréquence d'accès, durabilité, coûts
- Le client S3 peut indiquer la classe de stockage qu'il souhaite utiliser pour son objet

Mais il ne la précise pas toujours...



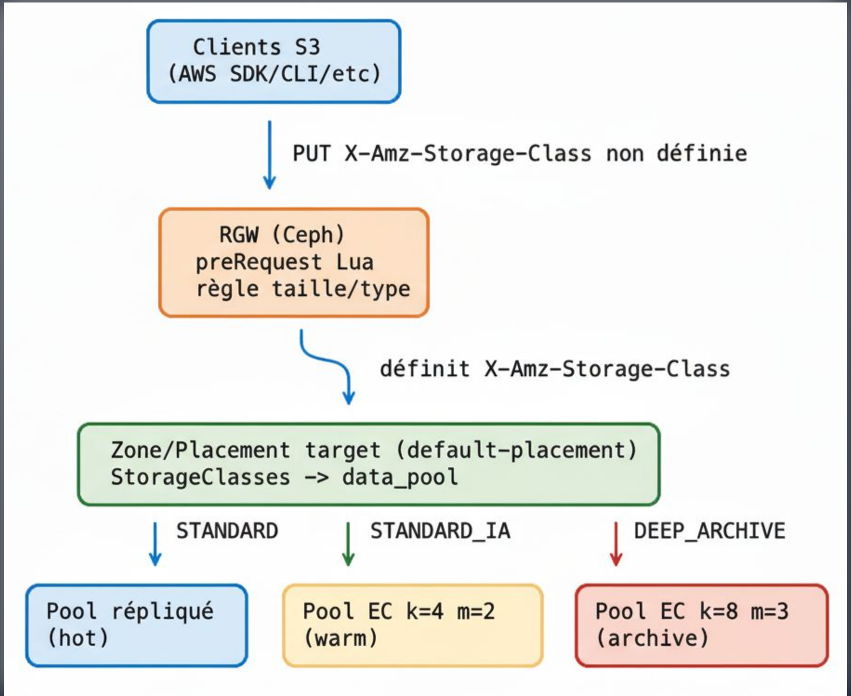
Placement dynamique

…et si on le faisait à sa place ?

- Lorsque le client ne la précise pas

- Affecter automatiquement une classe de stockage à un objet en fonction :
  - du type d'objet (.mp4, .data, .pdf)
  - de la taille de l'objet (< 64kB, > 1MB)
  - du tenant (labo1, labo2)
  - du bucket (gros ou petits objets)
  - de la méthode d'upload (MPU ou pas)

L'efficience de stockage et la performance d'accès seraient assurées dès l'écriture des données



Clients S3
(AWS SDK/CLI/etc)

PUT X-Amz-Storage-Class non définie

RGW (Ceph)
preRequest Lua
règle taille/type

définit X-Amz-Storage-Class

Zone/Placement target (default-placement)
StorageClasses -> data_pool

STANDARD

STANDARD_IA

DEEP_ARCHIVE

Pool répliqué
(hot)

Pool EC k=4 m=2
(warm)

Pool EC k=8 m=3
(archive)

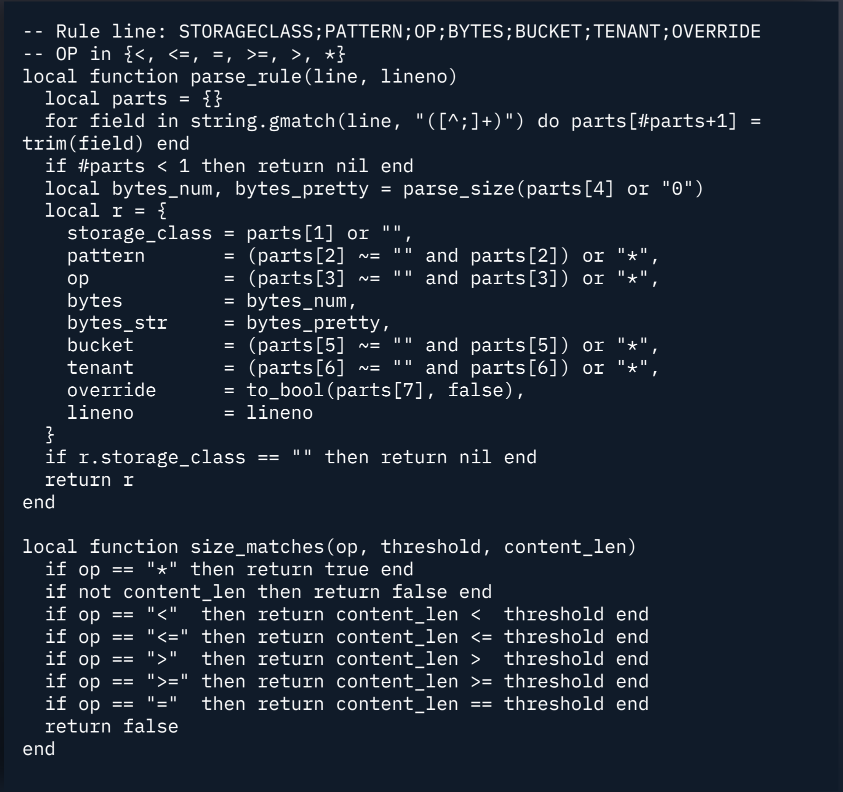
Placement dynamique - LUA Scripting

- Les passerelles S3 peuvent exécuter des scripts LUA à la volée en fonction du contexte (prerequest, postrequest, background, getdata, putdata)

- Ces scripts LUA peuvent lire et modifier les métadonnées d'un objet avant de l'écrire

  On peut donc dynamiquement définir ou modifier la classe de stockage d'un objet dès son écriture, selon certains critères

docs.ceph.com/en/latest/radosgw/lua-scripting



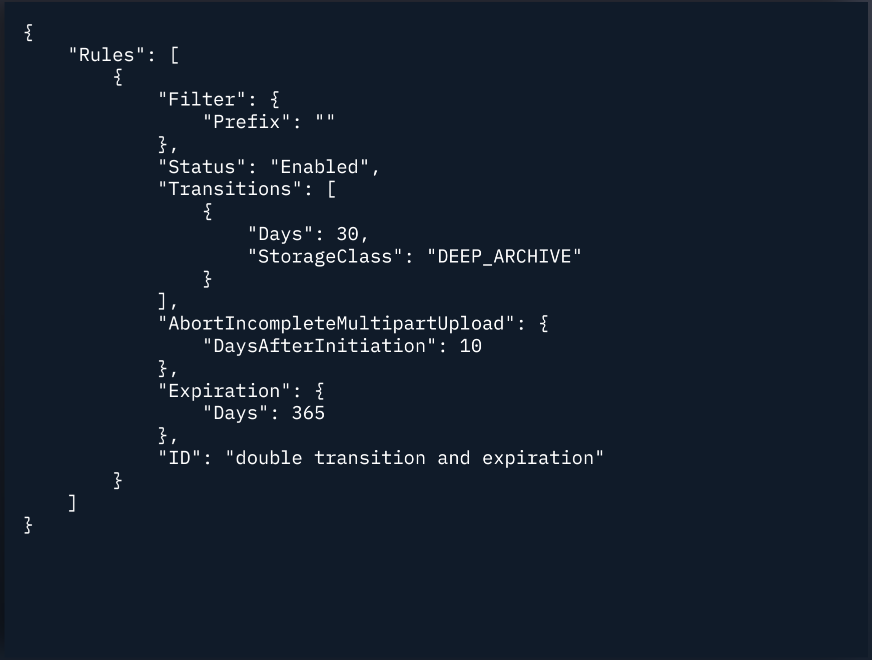
```
-- Rule line: STORAGECLASS;PATTERN;OP;BYTES;BUCKET;TENANT;OVERRIDE
-- OP in {<, <=, =, >=, >, *}
local function parse_rule(line, lineno)
  local parts = {}
  for field in string.gmatch(line, "([^;]+)") do parts[#parts+1] =
trim(field) end
  if #parts < 1 then return nil end
  local bytes_num, bytes_pretty = parse_size(parts[4] or "0")
  local r = {
    storage_class = parts[1] or "",
    pattern       = (parts[2] ~= "" and parts[2]) or "*",
    op            = (parts[3] ~= "" and parts[3]) or "*",
    bytes         = bytes_num,
    bytes_str     = bytes_pretty,
    bucket        = (parts[5] ~= "" and parts[5]) or "*",
    tenant        = (parts[6] ~= "" and parts[6]) or "*",
    override      = to_bool(parts[7], false),
    lineno        = lineno
  }
  if r.storage_class == "" then return nil end
  return r
end

local function size_matches(op, threshold, content_len)
  if op == "*" then return true end
  if not content_len then return false end
  if op == "<"  then return content_len <  threshold end
  if op == "<=" then return content_len <= threshold end
  if op == ">"  then return content_len >  threshold end
  if op == ">=" then return content_len >= threshold end
  if op == "="  then return content_len == threshold end
  return false
end
```

Rétention optimisée - Lifecycle Policies

- Les Lifecycle Policies assurent la transition d'objets S3 entre classes de stockage après un certain temps
- Les données les plus anciennes sont réécrites pour occuper moins de place dans le cluster

La règle ci-contre :

- ➢ nettoie les MPU parts à 10 jours
- ➢ déplace les objets vers DEEP_ARCHIVE mappée au pool EC 8+3 à 30 jours
- ➢ supprime les objets à 365 jours



```
{
    "Rules": [
        {
            "Filter": {
                "Prefix": ""
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 30,
                    "StorageClass": "DEEP_ARCHIVE"
                }
            ],
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 10
            },
            "Expiration": {
                "Days": 365
            },
            "ID": "double transition and expiration"
        }
    ]
}
```

docs.ceph.com/en/latest/radosgw/lua-scripting

DEMO

Placement dynamique et rétention optimisée

Configuration du service RGW (S3)

Creation des pools ceph

```
ceph osd erasure-code-profile set ec42 k=4 m=2
ceph osd erasure-code-profile set ec83 k=8 m=3

ceph osd pool create s3.hot.data 256 256 replicated
ceph osd pool create s3.warm.data 128 128 erasure ec42
ceph osd pool create s3.archive.data 64 64 erasure ec83

ceph osd pool application enable s3.hot.data rgw
ceph osd pool application enable s3.warm.data rgw
ceph osd pool application enable s3.archive.data rgw
```

Ajout des Storage Classes et Storage Class par défaut

```
radosgw-admin zonegroup placement add --rgw-zonegroup france
--placement-id default-placement --storage-class STANDARD_IA

radosgw-admin zonegroup placement add --rgw-zonegroup france
--placement-id default-placement --storage-class DEEP_ARCHIVE

radosgw-admin zonegroup placement default --rgw-zonegroup france
--placement-id default-placement --storage-class STANDARD_IA

radosgw-admin zonegroup get | grep default_placement
    "default_placement": "default-placement/STANDARD_IA"
```

```
radosgw-admin period update --commit
```

Vérification

```
radosgw-admin zone placement list
[
    {
        "key": "default-placement",
        "val": {
            "index_pool": "s3.buckets.index",
            "storage_classes": {
                "DEEP_ARCHIVE": {
                    "data_pool": "s3.archive.data",
                    "compression_type": "zstd"
                },
                "STANDARD": {
                    "data_pool": "s3.hot.data"
                },
                "STANDARD_IA": {
                    "data_pool": "s3.warm.data",
                    "compression_type": "lz4"
                }
            },
            "data_extra_pool": "s3.buckets.non-ec",
            "index_type": 0,
            "inline_data": true
        }
    }
]
```

Configuration du service RGW (S3)

Renseigner les data_pool (et index/non-ec) côté zone

```
# STANDARD est déjà présente; s'assurer qu'elle pointe vers le pool répliqué "hot"
radosgw-admin zone placement add --rgw-zone nancy \
  --placement-id default-placement \
  --storage-class STANDARD \
  --data-pool s3.hot.data \
  --index-pool s3.buckets.index \
  --data-extra-pool s3.buckets.non-ec

# STANDARD_IA -> EC 4+2
radosgw-admin zone placement add --rgw-zone nancy \
  --placement-id default-placement \
  --storage-class STANDARD_IA \
  --data-pool s3.warm.data \
  --compression lz4

# DEEP_ARCHIVE -> EC 8+3
radosgw-admin zone placement add --rgw-zone nancy \
  --placement-id default-placement \
  --storage-class DEEP_ARCHIVE \
  --data-pool s3.archive.data \
  --compression zstd
```

Appliquer la configuration

```
radosgw-admin period update --commit
```

Configuration du service RGW (S3)

Modifier la configuration du service RGW

```
$ ceph orch ls --export --service_type=rgw --format yaml > rgw.yaml

$ vim rgw.yaml
service_type: rgw
service_id: monde-france-nancy
service_name: rgw.monde-france-nancy
placement:
  count: 1
  label: rgws_nancy
spec:
  rgw_frontend_port: 8080
  rgw_realm: monde
  rgw_zone: nancy
  rgw_zonegroup: france
custom_configs:
  - mount_path: /etc/ceph/rgw_storageclass_rules.conf
    content: |
      # Paramètres globaux MPU
      mpu_default_class=DEEP_ARCHIVE
      mpu_force=true

      # Paramètres globaux défaut non-MPU
      default_class=STANDARD_IA
      default_force=false

      # Règles PUT objet (non-MPU)
      # STORAGECLASS;PATTERN;OP;BYTES;BUCKET;TENANT;OVERRIDE
      #STANDARD_IA;%.pdf;*;0;*;*;true
      #INTELLIGENT_TIERING;*;<;32768;bucket-logs;*;true
      #ONEZONE_IA;%.eml;*;0;*;tenant-a;false
      #GLACIER;%.iso;<;1073741824;media-bucket;*;true

      STANDARD;*;<=;2MiB;*;*;true
      DEEP_ARCHIVE;%.data;*;0B;*;*;true
```

Appliquer la configuration et redéployer les RGWs

```
$ ceph orch apply -i rgw.yaml
Scheduled rgw.monde-france-metz update...
Scheduled rgw.monde-france-nancy update…

$ ceph orch redeploy rgw.monde-france-nancy
Scheduled to redeploy rgw.monde-france-nancy.r03h01.veyyjq on host 'r03h01'
```

Vérification

```
$ container_id=$(ssh -n r03h01 podman ps | grep rgw | awk '{print $1}')

$ ssh -n r03h01 podman exec -it $container_id cat
/etc/ceph/rgw_storageclass_rules.conf
# MPU PUTs settings
mpu_default_class=DEEP_ARCHIVE
mpu_force=true

# non-MPU PUTs settings
default_class=STANDARD_IA
default_force=false

# non-MPU PUTs rules
# STORAGECLASS;PATTERN;OP;BYTES;BUCKET;TENANT;OVERRIDE
#STANDARD_IA;%.pdf;*;0;*;*;true
#INTELLIGENT_TIERING;*;<;32768;bucket-logs;*;true
#ONEZONE_IA;%.eml;*;0;*;tenant-a;false
#GLACIER;%.iso;<;1073741824;media-bucket;*;true

STANDARD;*;<=;2MiB;*;*;true
DEEP_ARCHIVE;%.data;*;0B;*;*;true
```

Configuration du service RGW (S3)

Téléchargement du script LUA github.com/frednass/s3-dynamic-placement-and-archiving

```
$ curl -k -s https://raw.githubusercontent.com/frednass/s3-dynamic-placement-and-archiving/refs/heads/main/rgw_storageclass_rules.lua -o rgw_storageclass_rules.lua

$ radosgw-admin script put --infile=./rgw_storageclass_rules.lua --context=preRequest
```

Ajout du script LUA à la Rados Gateway (RGW)

```
$ curl -k -s https://raw.githubusercontent.com/frednass/s3-dynamic-placement-and-archiving/refs/heads/main/rgw_storageclass_rules.lua -o rgw_storageclass_rules.lua

$ radosgw-admin script put --infile=./rgw_storageclass_rules.lua --context=preRequest
```

Application de la Lifecyle Policy S3

Création du bucket

```
$ rclone mkdir s3:/newbucket2
```

Créer la Lifecycle Policy

```
$ vim lifecycle.json
{
    "Rules": [
        {
            "Filter": {
                "Prefix": ""
            },
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 30,
                    "StorageClass": "DEEP_ARCHIVE"
                }
            ],
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 10
            },
            "Expiration": {
                "Days": 365
            },
            "ID": "double transition and expiration"
        }
    ]
}
```

Ajouter la Lifecycle Policy au bucket test

```
$ aws s3api put-bucket-lifecycle-configuration \
--endpoint http://10.38.1.59:8080 \
--bucket newbucket2 \
--lifecycle-configuration file://lifecycle.json
```

Vérifier son application

```
$ aws s3api --endpoint http://10.38.1.59:8080 get-bucket-lifecycle-configuration
--bucket newbucket2
{
    "Rules": [
        {
            "Expiration": {
                "Days": 365
            },
            "ID": "double transition and expiration",
            "Prefix": "",
            "Status": "Enabled",
            "Transitions": [
                {
                    "Days": 30,
                    "StorageClass": "DEEP_ARCHIVE"
                }
            ],
            "AbortIncompleteMultipartUpload": {
                "DaysAfterInitiation": 10
            }
        }
    ]
}
```

Application de la Lifecyle Policy S3

Vérifier sa prise en charge

```
$ radosgw-admin lc list
[
    {
        "bucket":
":newbucket2:f50809af-d67e-426c-bccd-78f8ff6af983.1276537.1",
        "shard": "lc.6",
        "started": "Thu, 01 Jan 1970 00:00:00 GMT",
        "status": "UNINITIAL"
    }
]
```

Déclencher manuellement son exécution

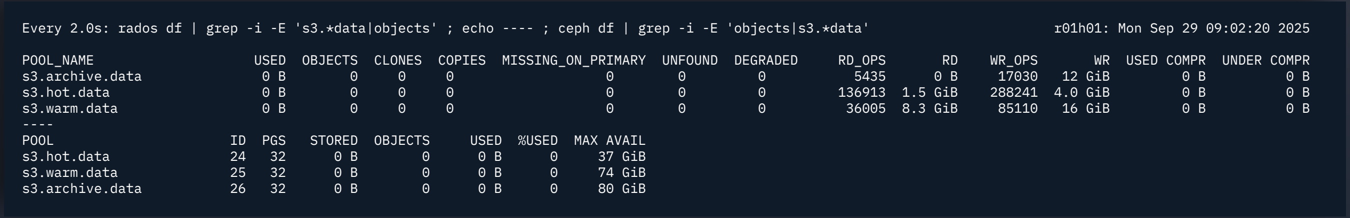
```
$ radosgw-admin lc process

$ radosgw-admin lc list
[
    {
        "bucket":
":newbucket2:f50809af-d67e-426c-bccd-78f8ff6af983.1276537.1",
        "shard": "lc.6",
        "started": "Sun, 28 Sep 2025 09:10:43 GMT",
        "status": "COMPLETE"
    }
]
```

Vérification du placement dynamique

Augmenter le niveau de debug

```
$ ceph config set global debug_rgw 20
```

Surveiller les pools



```
Every 2.0s: rados df | grep -i -E 's3.*data|objects' ; echo ---- ; ceph df | grep -i -E 'objects|s3.*data'                       r01h01: Mon Sep 29 09:02:20 2025

POOL_NAME                    USED  OBJECTS  CLONES  COPIES  MISSING_ON_PRIMARY  UNFOUND  DEGRADED     RD_OPS       RD    WR_OPS       WR  USED COMPR  UNDER COMPR
s3.archive.data               0 B        0       0       0                   0        0         0       5435      0 B     17030   12 GiB         0 B          0 B
s3.hot.data                   0 B        0       0       0                   0        0         0     136913  1.5 GiB    288241  4.0 GiB         0 B          0 B
s3.warm.data                  0 B        0       0       0                   0        0         0      36005  8.3 GiB     85110   16 GiB         0 B          0 B
----
POOL                      ID  PGS   STORED  OBJECTS     USED  %USED  MAX AVAIL
s3.hot.data               24   32      0 B        0      0 B      0     37 GiB
s3.warm.data              25   32      0 B        0      0 B      0     74 GiB
s3.archive.data           26   32      0 B        0      0 B      0     80 GiB
```

Créer des objets de différentes taille de 16KiB à 4,7 MiB

```
for i in {1..300} ; do dd if=/dev/urandom of=file_${i}.dd bs=16k count=${i} ; done
```

Pousser les objets vers S3

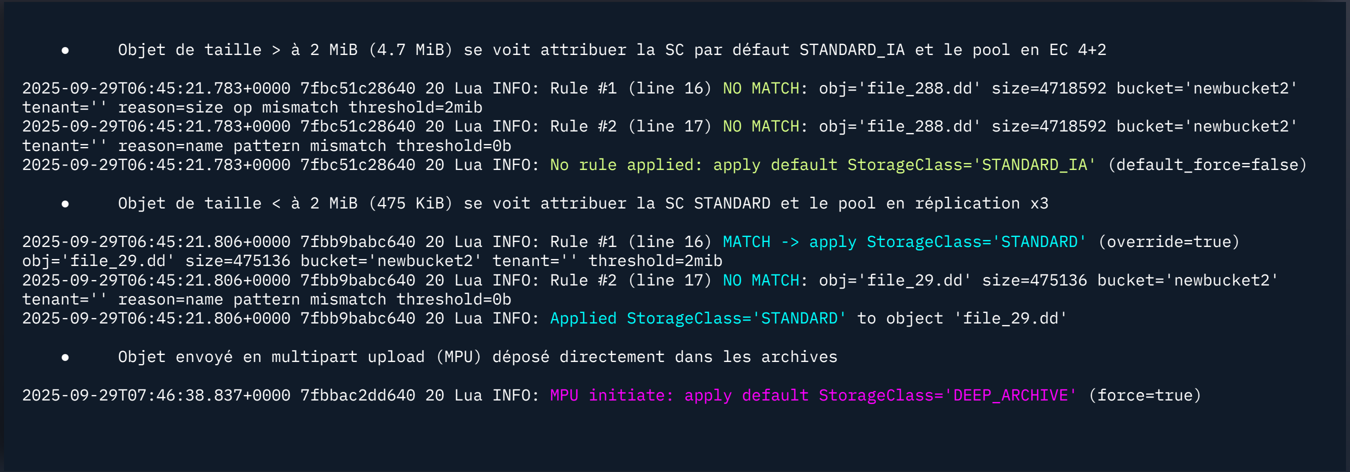
```
rclone sync . s3:/newbucket2 --s3-upload-cutoff 200M --s3-upload-concurrency 8
```

Contrôle des Storage Classes associées aux objets S3

```
radosgw-admin bucket list --bucket newbucket2
```

Vérification du placement dynamique

Journaux de la Rados Gateway (RGW)



- Objet de taille > à 2 MiB (4.7 MiB) se voit attribuer la SC par défaut STANDARD_IA et le pool en EC 4+2

```
2025-09-29T06:45:21.783+0000 7fbc51c28640 20 Lua INFO: Rule #1 (line 16) NO MATCH: obj='file_288.dd' size=4718592 bucket='newbucket2'
tenant='' reason=size op mismatch threshold=2mib
2025-09-29T06:45:21.783+0000 7fbc51c28640 20 Lua INFO: Rule #2 (line 17) NO MATCH: obj='file_288.dd' size=4718592 bucket='newbucket2'
tenant='' reason=name pattern mismatch threshold=0b
2025-09-29T06:45:21.783+0000 7fbc51c28640 20 Lua INFO: No rule applied: apply default StorageClass='STANDARD_IA' (default_force=false)
```

- Objet de taille < à 2 MiB (475 KiB) se voit attribuer la SC STANDARD et le pool en réplication x3

```
2025-09-29T06:45:21.806+0000 7fbb9babc640 20 Lua INFO: Rule #1 (line 16) MATCH -> apply StorageClass='STANDARD' (override=true)
obj='file_29.dd' size=475136 bucket='newbucket2' tenant='' threshold=2mib
2025-09-29T06:45:21.806+0000 7fbb9babc640 20 Lua INFO: Rule #2 (line 17) NO MATCH: obj='file_29.dd' size=475136 bucket='newbucket2'
tenant='' reason=name pattern mismatch threshold=0b
2025-09-29T06:45:21.806+0000 7fbb9babc640 20 Lua INFO: Applied StorageClass='STANDARD' to object 'file_29.dd'
```

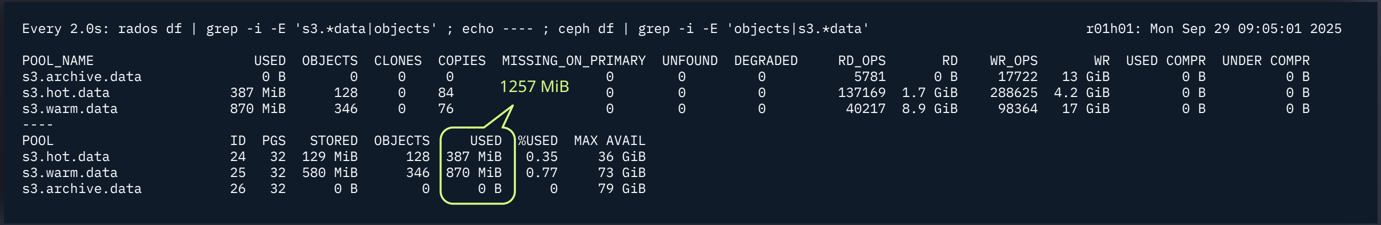
- Objet envoyé en multipart upload (MPU) déposé directement dans les archives

```
2025-09-29T07:46:38.837+0000 7fbbac2dd640 20 Lua INFO: MPU initiate: apply default StorageClass='DEEP_ARCHIVE' (force=true)
```

CL'ISO

Vérification de la rétention optimisée

Observer le remplissage des différents pools



```
Every 2.0s: rados df | grep -i -E 's3.*data|objects' ; echo ---- ; ceph df | grep -i -E 'objects|s3.*data'                    r01h01: Mon Sep 29 09:05:01 2025

POOL_NAME                  USED  OBJECTS  CLONES  COPIES  MISSING_ON_PRIMARY  UNFOUND  DEGRADED     RD_OPS       RD    WR_OPS       WR  USED COMPR  UNDER COMPR
s3.archive.data             0 B        0       0       0                   0        0         0       5781      0 B     17722   13 GiB         0 B          0 B
s3.hot.data             387 MiB      128       0      84                   0        0         0     137169  1.7 GiB    288625  4.2 GiB         0 B          0 B
s3.warm.data            870 MiB      346       0      76                   0        0         0      40217  8.9 GiB     98364   17 GiB         0 B          0 B
----
POOL                 ID  PGS   STORED  OBJECTS     USED  %USED  MAX AVAIL
s3.hot.data          24   32  129 MiB      128  387 MiB   0.35     36 GiB
s3.warm.data         25   32  580 MiB      346  870 MiB   0.77     73 GiB
s3.archive.data      26   32      0 B        0      0 B      0     79 GiB
```

1257 MiB

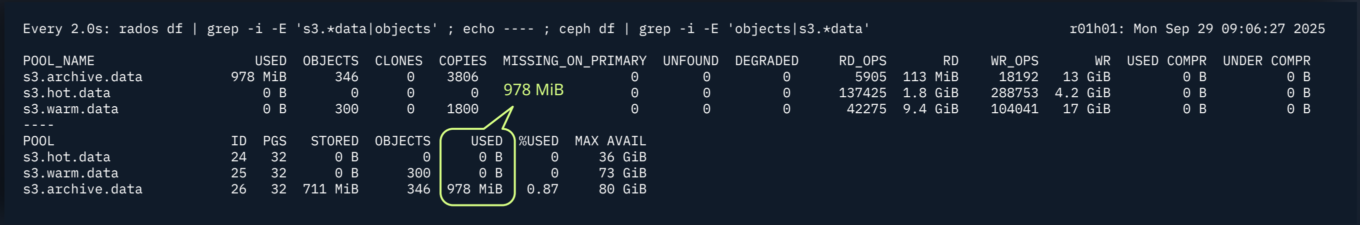
Réduire le temps entre chaque exécution

```
$ ceph config set global rgw_lc_debug_interval 1
```

Exécuter la Lifecycle Policy

```
$ radosgw-admin lc process
```

Nouveau contrôle (à 30s + radosgw-admin gc process --include-all)



```
Every 2.0s: rados df | grep -i -E 's3.*data|objects' ; echo ---- ; ceph df | grep -i -E 'objects|s3.*data'                    r01h01: Mon Sep 29 09:06:27 2025

POOL_NAME                  USED  OBJECTS  CLONES  COPIES  MISSING_ON_PRIMARY  UNFOUND  DEGRADED     RD_OPS       RD    WR_OPS       WR  USED COMPR  UNDER COMPR
s3.archive.data         978 MiB      346       0    3806                   0        0         0       5905  113 MiB     18192   13 GiB         0 B          0 B
s3.hot.data                 0 B        0       0       0                   0        0         0     137425  1.8 GiB    288753  4.2 GiB         0 B          0 B
s3.warm.data                0 B      300       0    1800                   0        0         0      42275  9.4 GiB    104041   17 GiB         0 B          0 B
----
POOL                 ID  PGS   STORED  OBJECTS     USED  %USED  MAX AVAIL
s3.hot.data          24   32      0 B        0      0 B      0     36 GiB
s3.warm.data         25   32      0 B      300      0 B      0     73 GiB
s3.archive.data      26   32  711 MiB      346  978 MiB   0.87     80 GiB
```

978 MiB

Nouveau contrôle (à 365s + radosgw-admin gc process --include-all) → les pools sont vides

Vérification de la rétention optimisée

Journaux de la Rados Gateway (RGW)

Transition à 15 et Expiration à 30 pour les tests

- Objet déposé

```
2025-09-29T08:12:56.866+0000 7fbc34bee640  1 beast: 0x7fbb4f01e6f0: 10.38.1.55 - fred [29/Sep/2025:08:12:56.842 +0000] "PUT /newbucket2/file_98.dd?x-id=PutObject HTTP/1.1"
200 1605632 - "rclone/v1.71.1" - latency=0.023999780s

2025-09-29T08:12:56.869+0000 7fbbf7373640  1 beast: 0x7fbb4f01e6f0: 10.38.1.55 - fred [29/Sep/2025:08:12:56.867 +0000] "HEAD /newbucket2/file_98.dd HTTP/1.1" 200 0 -
"rclone/v1.71.1" - latency=0.000999991s
```

- Objet transitionné vers la Storage Class DEEP_ARCHIVE

```
2025-09-29T08:13:11.052+0000 7fbc5bc5d640 20 lifecycle: operator()(): key=file_98.ddwp_thrd: 2, 0
2025-09-29T08:13:11.052+0000 7fbc5bc5d640 20 lifecycle: obj_has_expired(): mtime=2025-09-29T08:12:56.861730+0000 days=30 base_time=2025-09-29T08:13:11.053797+0000
timediff=15.0538 cmp=30 is_expired=0
2025-09-29T08:13:11.052+0000 7fbc5bc5d640 20 lifecycle: check(): key=file_98.dd: is_expired=0 wp_thrd: 2, 0
2025-09-29T08:13:11.052+0000 7fbc5bc5d640 20 lifecycle: obj_has_expired(): mtime=2025-09-29T08:12:56.861730+0000 days=15 base_time=2025-09-29T08:13:11.053827+0000
timediff=15.0538 cmp=15 is_expired=1
2025-09-29T08:13:11.052+0000 7fbc5bc5d640 20 lifecycle: check(): key=file_98.dd: is_expired=1 wp_thrd: 2, 02025-09-29T08:09:24.627+0000 7fbc5ac5b640  2 lifecycle:
TRANSITIONED::newbucket2[f50809af-d67e-426c-bccd-78f8ff6af983.1788835.1]):file_98.dd -> DEEP_ARCHIVE wp_thrd: 2, 2
```

- Objet supprimé

```
2025-09-29T08:13:26.005+0000 7fbc5d460640 20 lifecycle: operator()(): key=file_98.ddwp_thrd: 1, 1
2025-09-29T08:13:26.005+0000 7fbc5d460640 20 lifecycle: obj_has_expired(): mtime=2025-09-29T08:12:56.861730+0000 days=30 base_time=2025-09-29T08:13:26.005972+0000
timediff=30.006 cmp=30 is_expired=1
2025-09-29T08:13:26.005+0000 7fbc5d460640 20 lifecycle: check(): key=file_98.dd: is_expired=1 wp_thrd: 1, 1
2025-09-29T08:13:26.005+0000 7fbc5d460640 20 lifecycle: obj_has_expired(): mtime=2025-09-29T08:12:56.861730+0000 days=15 base_time=2025-09-29T08:13:26.005992+0000
timediff=30.006 cmp=15 is_expired=1
2025-09-29T08:13:26.005+0000 7fbc5d460640 20 lifecycle: check(): key=file_98.dd: is_expired=1 wp_thrd: 1, 1
2025-09-29T08:09:39.524+0000 7fbc5b45c640  2 lifecycle: DELETED::newbucket2[f50809af-d67e-426c-bccd-78f8ff6af983.1788835.1]):file_98.dd wp_thrd: 2, 1
```

Remerciements

- Yuval Lifshitz for adding Lua scripting into the Rados Object Gateway
- Steven Umbehocker (OSNEXUS) for his work on RGW autotiering that inspired this presentation
- Anthony D'Atri and Curt Bruns talk on RGW Lua scripting

Questions ?

Frédéric Nass
frederic.nass@clyso.com