Dossier Technique Scality RING

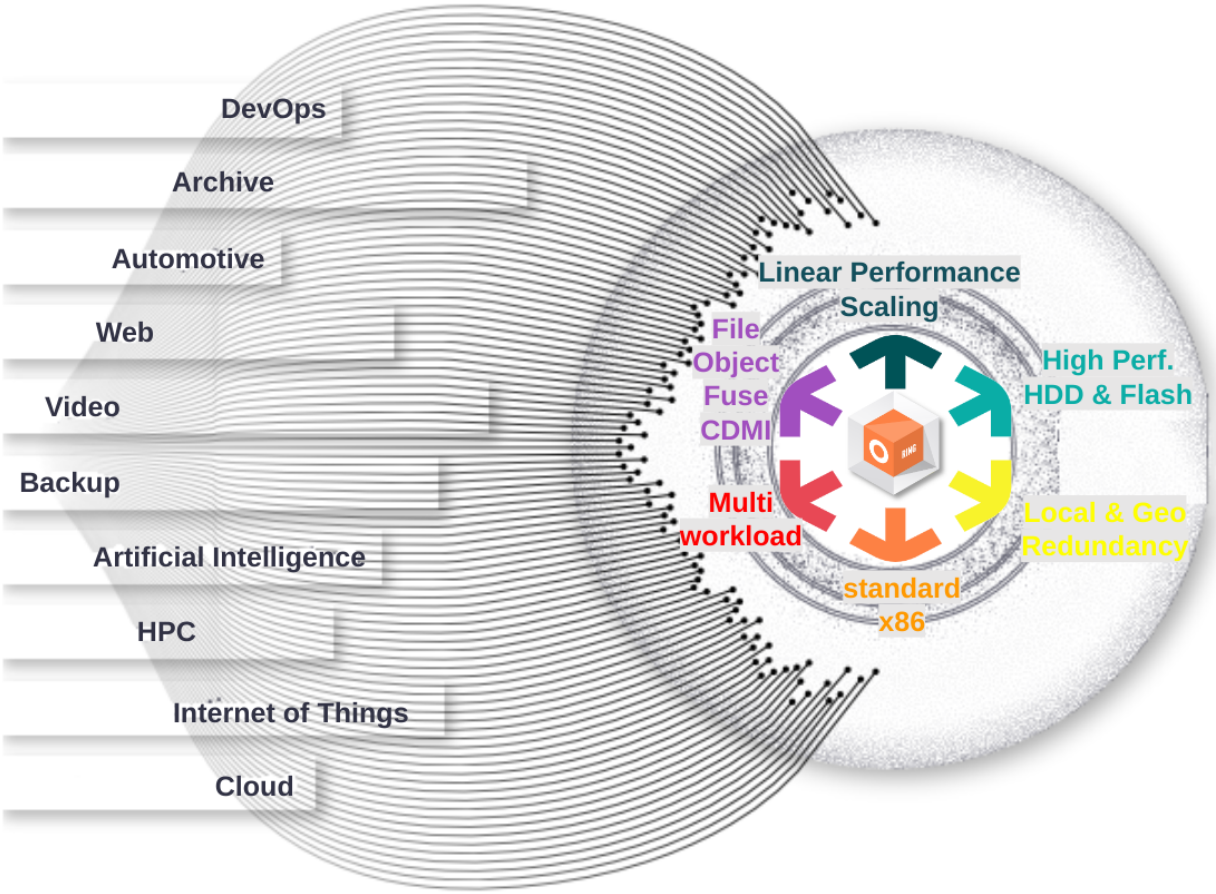
SCALITY

Table des matières

Présentation du RING Scality

Scality RING est le référentiel des data-lakes d'IA de choix pour les grandes entreprises des services financiers, de la génomique, des services publics, des services de renseignement, du voyage et des instituts de recherche. Parmi ses clients figurent : le laboratoire de génomique français SeqOIA Genomique, une des cinq plus grandes banques américaines pour la détection des fraudes, l'un des plus grands assureurs américains pour l'optimisation du traitement des sinistres, et le premier fournisseur européen de services de voyage basés sur l'IA.

Scality RING est une solution purement logicielle (**S**oftware **D**efined **S**torage) utilisant des serveurs standardisés et un système d'exploitation **Linux** comme socle. Il répond à une panoplie de cas d'usages provenant d'utilisations communes (intelligence artificielle avec les LLM, sauvegarde, archive, …) ou bien à différents segments verticaux (Santé, Industrie, Telco, Militaire, …).



La performance et l'évolution sont garanties par une architecture P2P (Peer-to-Peer) entièrement distribuée dont les performances croissent à mesure que le la capacité augmente. Au cœur du RING, une version étendue et brevetée de l'algorithme de P2P de seconde génération "Chord" a été insérée, ce qui lui confère une architecture entièrement distribuée sans point de contention (aucun SPOF). Ceci permet au RING de croître jusqu'à des capacités de plusieurs centaines de pétabytes.
Le RING exploite cette architecture en effectuant les opérations de lecture et écriture de façon massivement parallèle : sur les différents serveurs et sur tous les disques capacitifs en place. Comme le RING n'a pas de point de contention (ou SPOF), plus le nombre de disques et de serveurs est élevé, plus la performance augmente linéairement.

La disponibilité du système découle de cette architecture partagée, P2P et sans SPOF. La mise hors service d'un des éléments du RING (disque, serveur de stockage, connecteur, etc.) est immédiatement prise en compte par le RING qui, en déchargeant le matériel la responsabilité de la gestion des erreurs et des pannes, se charge de redistribuer la charge mais aussi la redondance sur les éléments restants. Le système sera par ailleurs dimensionné afin de garantir les performances même en cas de perte d'un serveur ou de tout autre composant matériel.

La durabilité des données est quant à elle assurée par l'utilisation extrêmement performante d'une combinaison de réplication et d'Erasure Coding, dont l'implémentation brevetée par Scality se nomme ARC. Son paramétrage flexible permet de répondre au plus près au cahier des charges, en maximisant durabilité et disponibilité des données d'une part, tout en optimisant le surcoût hardware d'autre part.

La simplicité opérationnelle du RING est représentée à travers différents aspects :

· En plus de son architecture scalable, le RING fournit les outils pour que les accroissements de capacité soient menés à bien de manière rapide et sans impact sur le trafic de production. L'extension de capacité se fait – toujours à chaud et sans interruption de service - par l'ajout de nouveaux serveurs de stockage, ceux-ci pouvant être remplis ou partiellement remplis de disques. La capacité utile fournie par le RING s'ajuste donc au mieux aux besoins réels de capacité.

· Dans la gestion du cycle de vie de ses données, comparé à une librairie LTO par exemple, le RING ne requiert à aucun moment de migrer d'une génération de bande à une autre, pas plus qu'il ne requiert de vérifier l'intégrité des supports. Ceci avec des temps d'accès et de débits en lecture et en écriture comparables aux NAS les plus rapides.

· Fonctionnant sur des serveurs x86, un OS Linux et un réseau 10GbE standards, la vie du RING se déroule sans interruption de service, en faisant coexister plusieurs générations de hardware, plusieurs fournisseurs si besoin, au fil des extensions de capacité et des cycles de support des serveurs, en passant d'une version du logiciel à la suivante, permettant ainsi de toujours bénéficier des dernières évolutions technologiques.

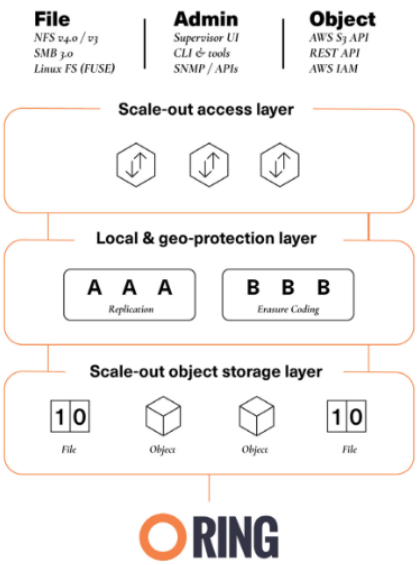
Les composants de la solution RING

Le RING de Scality est composé des éléments suivants :

- **Superviseur** - Il s'agit du serveur de gestion. Un Superviseur est requis par RING, mais un client multi-geo peut souhaiter avoir un Superviseur standby disponible pour le reconstruire en cas d'échec. Le Superviseur est le plus souvent installé sur une machine virtuelle.
- **Serveurs de stockage** - il s'agit du noyau du système, stockant les données et hébergeant les connecteurs. Un minimum de six serveurs de stockage par RING est requis. La croissance en étapes de trois serveurs de stockage par site est la norme.
- **Connecteurs** - Ils sont généralement installés directement sur les serveurs de stockage - ne nécessitant aucun matériel supplémentaire. Les machines physiques ou virtuelles externes peuvent être utilisées en option lors du support de protocoles multiples ou de réseaux segmentés.

La technologie RING est une solution purement SDS (Software Defined Storage). De ce fait, l'architecture Scality RING est complètement modulaire et agnostique vis à vis des châssis x86. Ainsi il est possible de concevoir un cluster RING à base d'élément d'un constructeur et intégrer par la suite

d'autres nœuds au sein du cluster à base d'un constructeur différent.
RING débute à partir de 3 nœuds (1 site) et évolue à plusieurs centaines de nœuds couvrant plusieurs sites à la fois.
Puisque les besoins de nos clients évoluent en architecture, en capacité ou en performance, le cluster RING évolue au fil de l'eau en complément des besoins. A titre d'exemple, nous pouvons citer le changement d'architecture : passage d'un cluster géo-stretched sur 2 sites à géo-stretched sur 3 sites ou alors l'ajout de connecteurs en mode découplés pour davantage de performances.
Cette modularité garantit la pérennité de l'architecture et de la solution et permet d'intégrer des technologies actuelles et nouvelles que ce soit sur les éléments matériels des noeuds, sur les technologies de disques ou bien sur le réseau : **cuivre, fibre, twinnax, 10GbE, 40GbE, 100GbE**, etc.

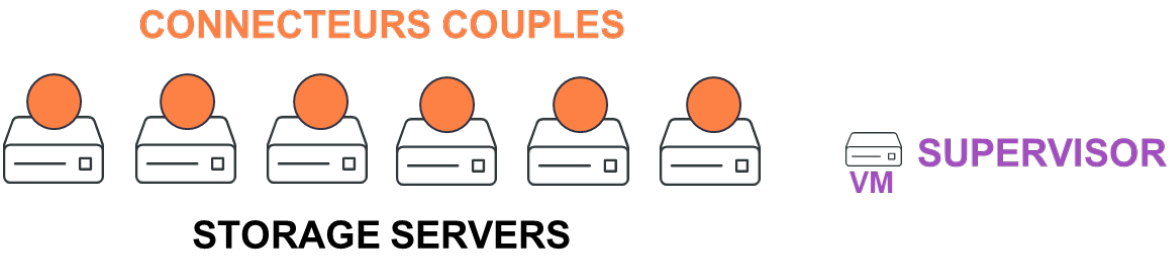


La solution Scality RING dispose d'une architecture "flex & multi-protocols" lui permettant des designs "couplés" ou bien "découplés". C'est à dire que l'on peut conserver les connecteurs de protocoles au sein des noeuds de stockage (couplés) ou bien les avoir à l'extérieur des noeuds de stockage (découplés) pour des raisons de sécurités (PROD <--> DMZ, tenants différents, ...) ou bien pour des raisons de performances. En effet, en multipliant les connecteurs il est ainsi possible d'augmenter la partie "compute" sans toutefois augmenter la partie "stockage" de l'architecture de stockage objet.
Scality RING dispose de connecteurs variés et multiples dont : Ring XP, S3, NFS et CIFS. Les connecteurs découplés peuvent être mis en place dès l'installation du RING ou bien par la suite pour un protocole ou un autre. Cela permet d'être évolutif et serein y compris pour de nouveaux protocoles.

Connecteurs : couplés

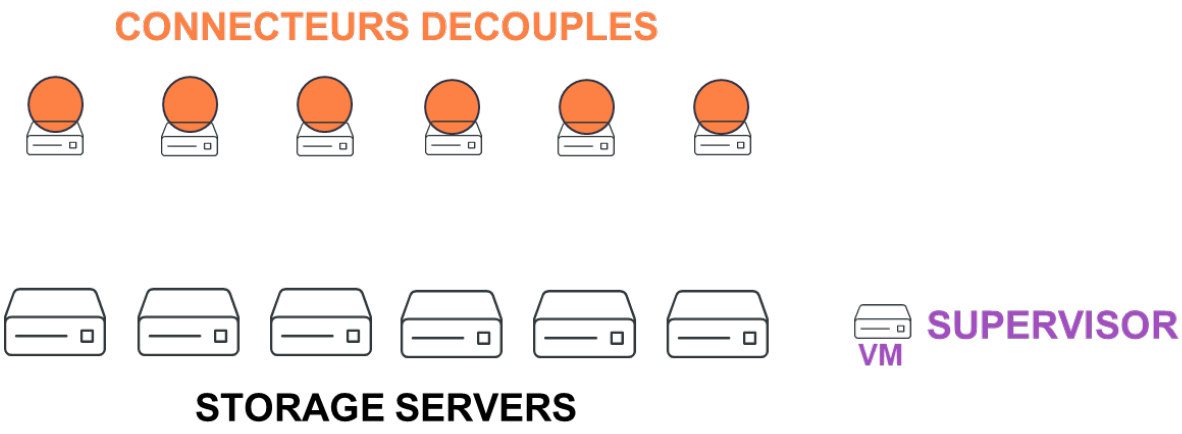
Comme évoqué lors de nos échanges, les architectures Scality RING sont granulaires et acceptent d'être couplés ou découplées. C'est-à-dire de conserver les connecteurs sur les nœuds RING (« storage server ») ou bien de les installer sur des serveurs différents des nœuds RING.
Il est ainsi possible d'optimiser son architecture initialement ou bien par la suite lors des évolutions futures afin de fournir, par exemple, un gain en performance sur un type de connecteur.

Voici un exemple de connecteurs (tous types) couplés sur les storage servers RING :



Connecteurs : découplés

Et sa version découplée sur un matériel différent (autre châssis HW ou bien VM) :


CONNECTEURS DECOUPLES
SUPERVISOR
VM
STORAGE SERVERS

Mise en réseaux

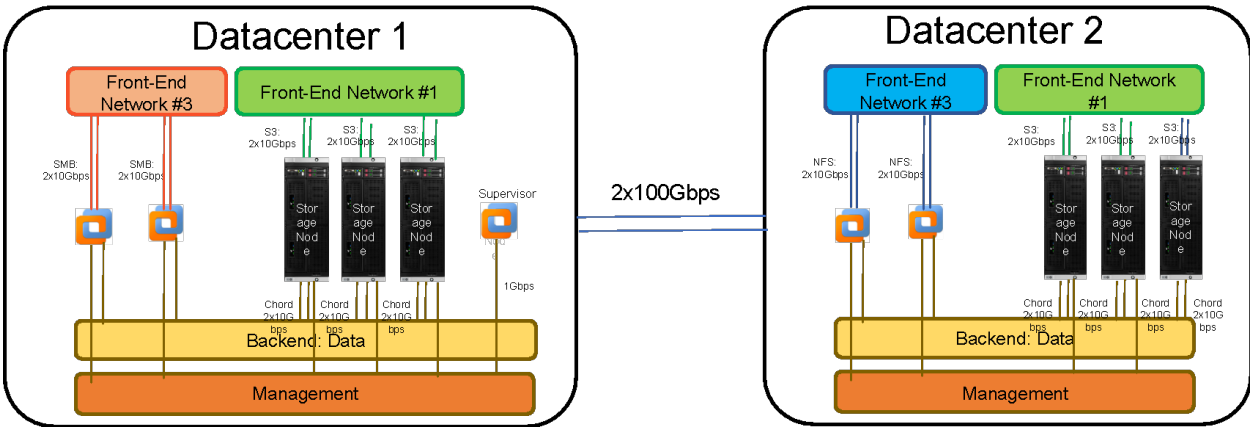
La séparation (ou étanchéité) entre les différents éléments quels qu'ils soient s'effectue en fonction de l'architecture choisie et du mode sélectionné : couplé ou découplé pour la partie réseau avec un ou plusieurs réseaux. Il existe plusieurs réseaux "de base" sur RING :

- **Back-End** : appelé aussi "data network" constitue le réseau de communication entre les nœuds de stockage mais également entre les connecteurs et les nœuds de stockage.
- **Front-End** : appelé aussi "application network" constitue le réseau de communication entre les applications clientes et les connecteurs
- **Management** : constitue le réseau de communication entre le superviseur, les connecteurs et les nœuds de stockage.

Les connecteurs peuvent être configurés dans des réseaux différents. Il est commun d'avoir un ou plusieurs réseaux frontends séparés du réseau backend. Les connecteurs doivent pouvoir communiquer avec les serveurs de stockage mais ne communiquent pas entre eux. Il est ainsi possible d'avoir des connecteurs configurés dans une DMZ, tandis que d'autres sont dans une zone « interne ».

Ci-dessous un schéma montrant quelques possibilités d'intégration d'un RING dans un réseau client. Dans cet exemple, nous avons 2 connecteurs NFS installés sur des machines virtuelles dans le réseau « Front-end Network#3 », 2 connecteurs NFS installés sur les serveurs de stockage dans le réseau « Frontend Network#1 » et 4 connecteurs SMB et FTP dans le réseau « Frontend Network #2 ». L'ensemble de ces machines sont également dans le réseau de Backend (via une seconde carte réseau pour les serveurs de stockage par exemple, sinon via du Trunk).
Plusieurs options sont possibles :

- avoir des réseaux dissociés et un réseau de management séparé
- tout mettre dans un seul et même réseau.



Le nombre et le type d'interface réseau peut varier en fonction des besoins de nos clients. De part notre technologie SDS, nous sommes agnostiques au matériel et par conséquent nous pouvons opter pour le meilleur châssis x86 disposant des bus ou interfaces nécessaires à vos besoins. Par défaut, nous préférons avoir au moins 2 x 10 GbE (10Gb/25Gb en standard aujourd'hui dans les serveurs) afin de créer un bonding permettant la redondance et l'agrégation des interfaces réseaux.

Concernant la fonctionnalité "multi-tenant", Scality RING prend en charge le modèle de gestion multi-tenant au travers du connecteurs S3 et la gestion de compte standard AWS mais aussi via IAM (Identity Access Management) procurant l'accès à plusieurs comptes, utilisateurs et groupes.

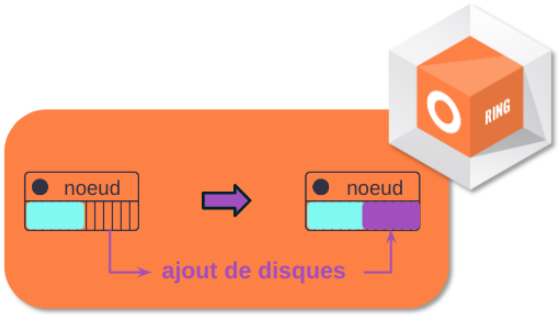
Les comptes (S3 root) et les utilisateurs peuvent avoir chacun leur propre accès - paire de clés d'accès et de clé secrète - utilisées pour l'authentification sécurisée via le schéma d'authentification basé sur HMAC AWS Signature v4 (et v2). Les stratégies IAM (policy) peuvent être affectées aux utilisateurs et aux groupes pour un accès très granulaire de contrôle (refuser / autoriser les privilèges).

Evolutions et Géo-répartition

La technologie RING dispose de tous les arguments pour évoluer nativement en mode "scale-IN" et en mode "scale-OUT" afin d'accroître la capacité du cluster ou bien la protection des données de nos clients avec la géo-répartition sur plusieurs datacenters.

Scale-In

En effet, il est possible d'ajouter un ou plusieurs disques internes (emplacement vide à l'origine dans les châssis) à chaque nœud RING permettant une évolution dite "**scale-IN**" sans contraindre nos clients à ajouter des nœuds complémentaires. Cette capacité est prise en compte dans la foulée et s'effectue sans interruption de service.



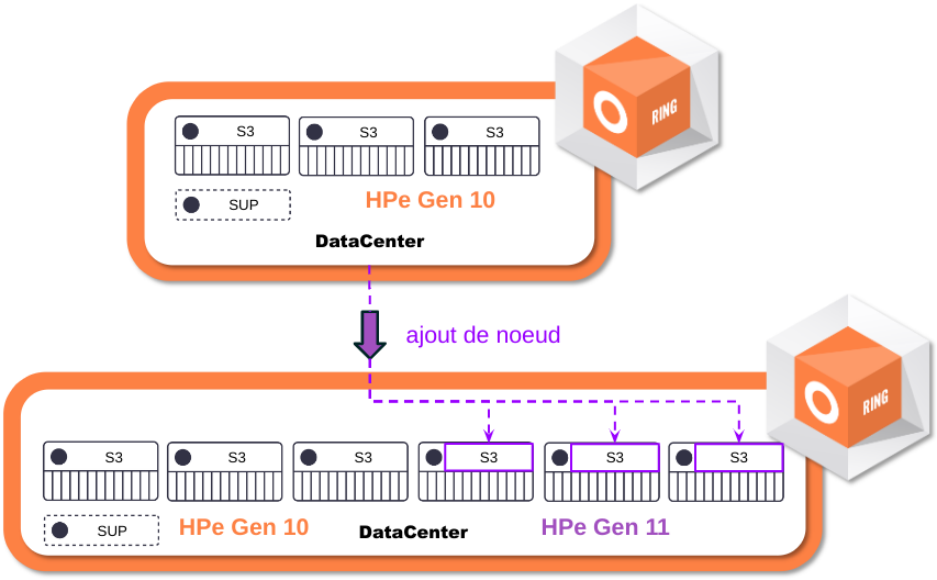
Dans un même cluster homogène à l'origine, les ajouts de disques sont réalisés de façon identique sur la totalité des nœuds du cluster.
L'ajout unitaire ou en grappe dépend de l'espace disponible restant dans le cluster/nœud et de la capacité des nouveaux disques. Un nœud RING peut accueillir des capacités de disques différentes. Il est recommandé d'évoluer à la hausse sur les capacités (et non à la baisse) : 10 To →14 To par exemple. Suivant la stratégie adoptée durant le design de l'architecture, l'évolution "scale-IN" dans un cluster peut être assez conséquente. En effet, les châssis peuvent comporter quelques slots vides (de l'ordre d'une dizaine) ou bien plus. Il existe des avantages et des inconvénients pour les deux solutions ; ils sont d'ordre technique (nombre de RU supérieur pour un châssis dense) et financier (coût d'entrée plus élevé mais plus faible pour les add-ons). Voici deux exemples pour illustrer et différencier les stratégies de réponse :

- châssis standard de 24 slots dont 12 remplis et **12 disponibles** →augmentation par disque et par grappe = on double la capacité par nœud au terme final.
- châssis dense de 60 slots dont 12 remplis et **48 disponibles** →augmentation par disque et par grappe = on obtient un facteur x5 par rapport à l'empreinte de base au terme final.

Le RING dispose d'un outil (*scaldisk*) permettant d'effectuer un "dry run" avant d'effectuer toutes opérations complémentaires sur la Production.

Scale-Out

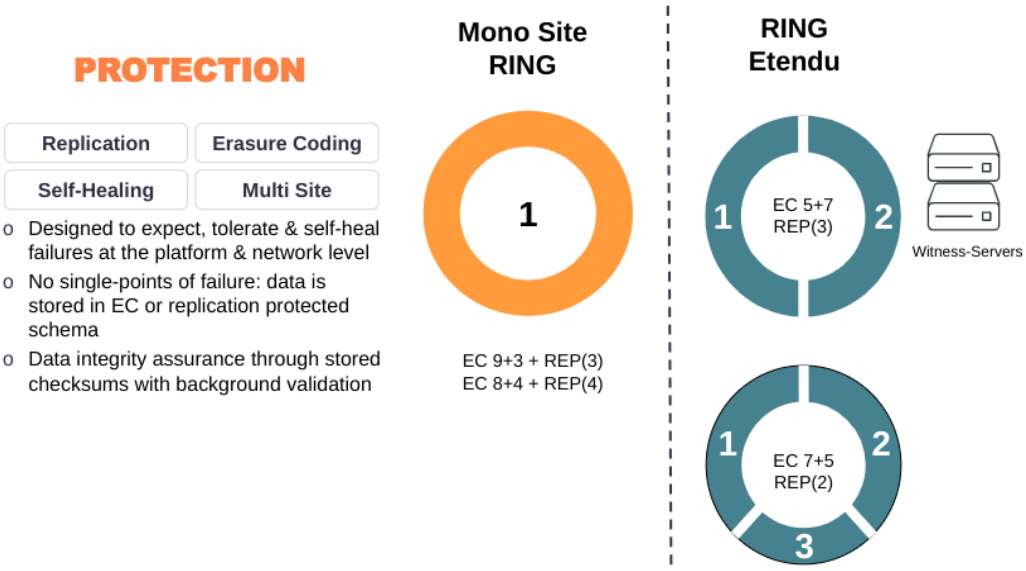
Il est aussi possible d'ajouter des nœuds complémentaires dans le cluster s'appuyant sur le mode "**scale-OUT**" pour accroître la capacité du cluster et la partie compute des storage nodes. Cette capacité est prise en compte dans la foulée et s'effectue sans interruption de service.



Dans l'éventualité d'une modification de marché, de contrat ou le désir de changer de type de châssis x86 ou tout simplement de fournisseur, sachez que le RING supporte les nœuds hétérogènes et de générations différentes. Le schéma ci-dessus illustre bien le support de génération différente au sein d'un même cluster RING. C'est déjà le cas au travers de nos partenaires constructeurs offrant de nouvelle génération et intégrant les nouveaux composants : CPU, carte réseau, technologie de disque, etc.

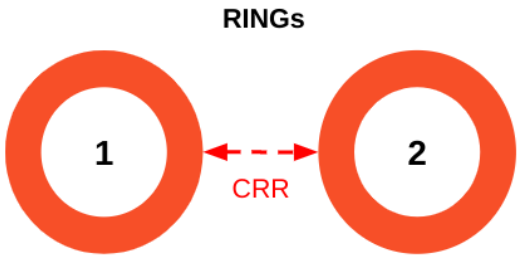
Géo-répartition

Ces compléments de disques ou de nœuds s'accompagnent bien souvent d'une mise en disponibilité des données sur plusieurs sites permettant d'être résilient à la perte de disques, de nœuds mais aussi d'un rack, d'une salle ou bien d'un site complet.
Finalement, les architectures RING sont diverses et variées : 1 site, 2 sites stretched ou miroir ou 3 sites stretched, ...



La réplication des données au sein du RING est mise en place entre deux systèmes par la fonctionnalité

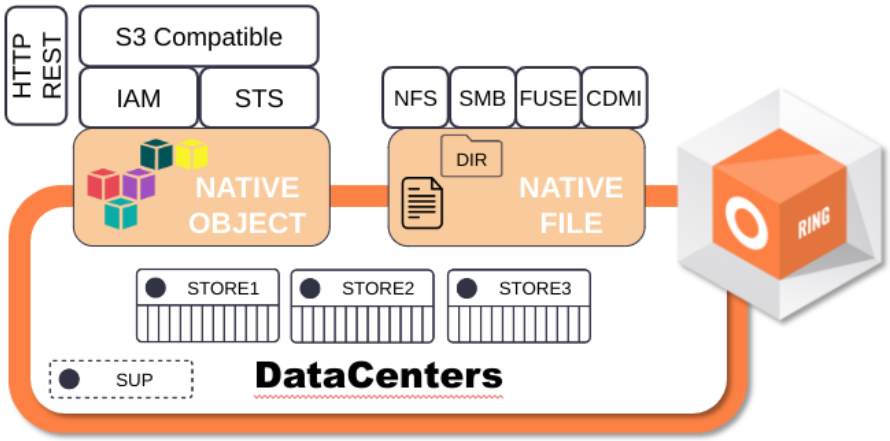
CRR (Cross Region Replication). Elle s'effectue au niveau des buckets et réplique les objets d'y trouvant.



L'administrateur décide des buckets à répliquer et applique une réplication de type "bucket source" vers "bucket destination".
La fonctionnalité CRR nécessite d'activer la fonctionnalité "versioning" au sein des buckets source et destination.
Répliquer la totalité d'un cluster, il s'agit d'une architecture de type "mirrored RINGs" mettant en œuvre une protection locale (COS & ARC) et une protection distante de type Disaster Recovery par exemple.

Multi-Protocoles

Le RING Scality dispose nativement de nombreuses interfaces de communication (nommées : “connecteurs”) pour accéder au stockage des données.



Les connecteurs disponibles à ce jour sont :

Type	Connecteur	Fonctionnalité
Objet	S3	Interface hautement compatible AWS S3, prise en charge d’AWS IAM v2 et v4 ainsi qu’ActiveDirectory.
	SWIFT	Interface objet d’OpenStack Swift
	Ring XP	Connecteur objet eXtreme Performance dédié à l’IA avec une latence TTFB inférieure à 1ms
	HTTP REST	API REST Scality native et légère
Fichier	NFS	NFSv3 et NFSv4, prise en charge de Kerberos.
	SMB	SMBv2 et un sous-ensemble de SMBv3
	FUSE	Pilote de système de fichiers Linux local, idéal pour les serveurs d'applications
	CDMI	Interface standard SNIA, accès REST au fichier.

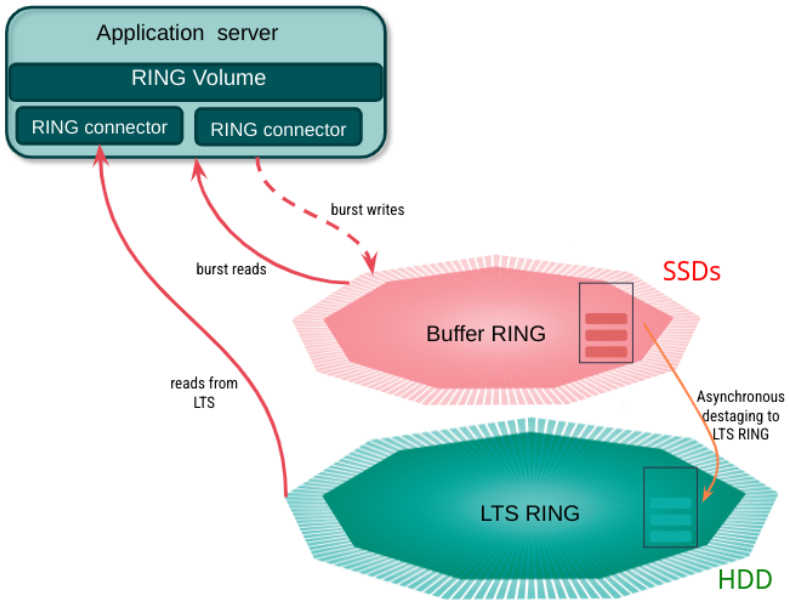
Le système de fichier

Entièrement redondé et basé sur de multiples accès parallèles aux disques (HDD ou SSD), le RING fournit un système POSIX reposant sur une base de donnée interne de type NoSQL qui peut s’étendre à l’infini. Ce système permet une gestion en mode couplés ou découplés autorisant la redondance et l’évolution à de multiples connecteurs de type “Fichiers”. Les fonctionnalités en standard sont :

- load balancing fichiers intégré,
- gestion des quotas,
- File undelete (corbeille) et versioning,
- Volume protection (WORM),
- NFS v3 et v4 avec authentification Kerberos et support IPV6,
- SMB v2 et v3,
- Linux FUSE,
- Storage Accelerator pour la mise en cache

Principe du « Storage Accelerator SOFS »

L'objectif de la fonctionnalité « Storage Accelerator SOFS » est l'optimisation des écritures et des lectures provenant de vos applications. En effet, elles vont pouvoir bénéficier d'un cache fournissant les performances supplémentaires pour une période de rétention correspondant à la période d'utilisation de ces données. Ceci est bien pratique pour des documents devant être « boostés » sur quelques heures par exemple pour effectuer des examens ou un travail momentané.
Par la suite, ces données seront déplacées automatiquement sur un emplacement de protection dit « à long terme » et de nature plutôt capacitive.



Le délai de conservation en cache (ou déplacement sur le socle long terme) est configurable en fonction des besoins. Il doit toutefois être pris en compte dès l'origine afin de fournir les bonnes capacités de disques SSDs (sous « standard SSD » dans le sizing).

Le buffer (cache) se désengage automatiquement si la capacité venait à manquer afin de garantir les écritures y compris sur le socle à long terme au détriment de la performance.

API S3 et IAM

Le RING Scality comprend une API S3 hautement compatible avec celle développée par AWS. Étant donné qu'il s'agit de la norme de référence pour le stockage d'objets, Scality a investi dans la recherche et le développement pour s'assurer qu'il s'agit d'une véritable représentation de l'API S3 d'AWS. Au

fur et à mesure qu'AWS met en place de nouvelles fonctionnalités, Scality cherche à les mettre en œuvre rapidement grâce à la méthodologie de développement agile du RING. Comme l'API S3 de Scality suit les mêmes normes que l'API S3 d'AWS, l'intégration, avec les ISVs compatibles S3, fonctionne comme prévu.

La liste précise des APIs S3 supportées par Scality se trouve dans le guide de référence S3 de Scality (S3 Connector Reference).

En complément de l'API S3, Scality a implémenté les autres API également : IAM (Identity Access Management) ou alors STS (Security Token Service).

Par ailleurs, Scality utilise et supporte cette méthodologie API pour étendre les fonctionnalités en mode REST API pour des fonctionnalités telles que le reporting et les métriques pour un tenant via UTAPI (UTilization API).

Ring XP

Avec Ring XP, Scality accélère le stockage objet pour l'IA : Capable de descendre à 500 microsecondes de latence sur des SSD NVMe, la nouvelle déclinaison du système Ring devient comparable aux systèmes de fichiers utilisés en IA, avec l'avantage supplémentaire de ne pas avoir de limite de taille.

RING XP surpasse les performances d'Amazon S3 Express One Zone. Il s'agit de la première solution de stockage objet software-defined à atteindre de tels niveaux de performance – jusqu'alors réservés aux systèmes de fichiers et baies de stockage SAN 100 % flash – tout en offrant les avantages inhérents au stockage objet en termes d'évolutivité, de simplicité, d'accès API, de sécurité et de coût.

Pour porter les performances du stockage objet au niveau supérieur avec des latences de l'ordre de la microseconde, RING XP s'appuie sur :

- Des connecteurs de stockage objet RING XP optimisés pour l'IA afin de fournir un accès scale-out et rapide au stockage depuis les applications ;
- Un logiciel de stockage RING optimisé pour les performances qui accélère les Entrées/Sorties à tous les niveaux ;
- Des serveurs de stockage NVMe 100 % flash basés sur AMD EPYC™ de Lenovo, Supermicro, Dell et HPE. EPYC offre une prise en charge PCIe et NVMe leader du secteur, ainsi que le plus grand nombre de cœurs dans les processeurs pour des latences optimales.

En atteignant une latence d'écriture (PUT) et de lecture (GET) de l'ordre de la microseconde pour des objets de 4 Ko, RING XP offre des performances de stockage objet ultra-rapide idéale pour les outils d'IA, les applications développées sur mesure et les systèmes de fichiers optimisés pour les performances utilisés pour l'entraînement des modèles d'IA.

RING XP, associé à RING, offre une solution complète de gestion du stockage des pipelines de données d'IA, conçue pour optimiser et accélérer les processus métier émergents basés sur l'IA. Contrairement aux solutions de stockage traditionnelles qui ne traitent que des fragments du workflow de l'IA, RING XP et RING fournissent une plateforme unifiée qui prend en charge chaque étape du pipeline de l'IA, de l'ingestion de jeux de données massifs à l'entraînement des modèles, en passant par l'inférence et au-delà.

Les capacités de l'implémentation S3 dans RING

Tant les performances que les fonctionnalités S3 sont améliorées à chaque nouvelle release de notre logiciel RING. Voici les dernières tables disponibles dans notre documentation officielle pour RING 9.4 (documentation "S3 connector" en annexe).

Total Buckets in Metadata Cluster | **Number of Objects per Bucket**

Features	100K+	1M+	100M+	1B+
Basic Storage Operation	Supported	Supported	Supported	Supported
CRR	Supported	Supported	Supported	Consult Scality
Bucket Notifications	Supported	Supported	Supported	Consult Scality
Lifecycle Expiration	Supported	Supported	Supported	Consult Scality
Quotas Powered by Utapi V2	Supported	Supported	Supported	Consult Scality
Utapi V2	Supported	Supported	Supported	Consult Scality

S3 API Capabilities | IAM Capabilities

S3 API Description Capabilities	Amazon	S3 Connector
Number of accounts	unlimited	unlimited (depending on hardware)
User Metadata	2 KB	2 KB
Number of buckets	100 (extendable to 1000)	1000+ per account
Number of objects per bucket	unlimited	unlimited (depending on hardware)
Number of empty objects read per sec	300	1500 per cluster (depending on hardware)
Number of empty objects written per sec	100	3000 per cluster (depending on hardware)
# Tags per object	10	10
Key size for a tag	128	128
Value size for a tag	256	256
Number of lifecycle rules per account.	1000	1000

Note: With *1000*+ the + means there's no hard limit. Scality follows the Amazon limitations for compatibility. The system will accept any creation above this number.

Le MPU (Multipart Upload) dans le protocole S3 offre une méthode efficace et robuste pour uploader de gros objets en les divisant en parties plus petites (parts), qui sont téléchargées indépendamment et parallèlement.

Limits for the size of objects	
Maximum size of object if not uploaded as MPU	No limit
Maximum size of object if uploaded as MPU	50 TB
Minimum part size limit for MPU	5 MB
Maximum part size limit for MPU	5 GB
Minimum number of parts for MPU	1
Maximum number of parts for MPU	10000

Ainsi, les objets sont divisés en parties plus petites lorsque c'est possible et nécessaire (jusqu'à 10 000 parties), ce qui facilite le téléchargement atteignant plusieurs mégaoctets, gigaoctets et plus. Les avantages sont les suivants :

- Résilience aux échecs

Avec MPU, en cas de panne réseau ou d'erreur, seules les parties échouées doivent être retransmises. Il n'est pas nécessaire de recommencer tout l'upload. Cela améliore considérablement la fiabilité, notamment dans des environnements où les connexions réseau peuvent être instables.

- Performance optimisée (transferts en parallèles)

Le protocole MPU permet d'uploader plusieurs parties en parallèle, ce qui accélère considérablement le transfert total. Cela utilise de manière optimale la bande passante disponible. Par exemple, un fichier de 10 Go peut être découpé en 10 parties de 1 Go chacune, transférées simultanément sur plusieurs connexions.

- Reprise des uploads interrompus

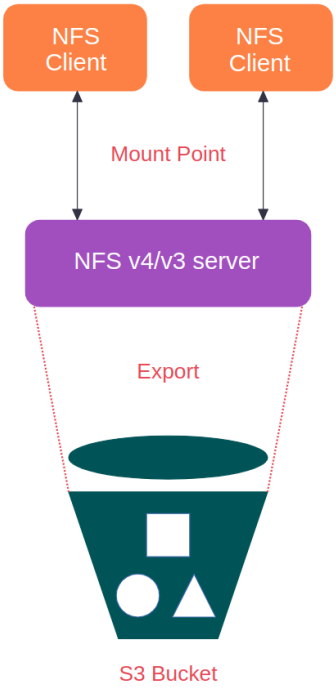
Le MPU permet de reprendre un téléchargement interrompu là où il s'était arrêté, sans avoir à recommencer depuis le début. Cette fonctionnalité est particulièrement utile pour les objets très volumineux. Le “partial upload” S3 est stocké durant une période de temps permettant de compléter l'upload ultérieurement.

En standard, le fonctionnement est le suivant :

1. Initiation : Une requête d'initialisation démarre l'upload multipart, générant un identifiant unique pour cet upload.
2. Division en parties : Le fichier est découpé côté client en morceaux plus petits (généralement entre 5 Mo et 5 Go chacun).
3. Upload des parties : Les parties sont envoyées individuellement (parallèlement ou séquentiellement).
4. Assemblage final : Une fois toutes les parties envoyées, une requête de finalisation assemble les parties dans l'objet final.

Cross protocole

Pour les données nécessitant d'être accéder par S3 et un autre protocole de type "file", il est possible de le faire au travers du protocole NFS. Cette méthodologie a été mise en place pour faciliter les migrations d'application de type "legacy" vers le protocole S3. Il faut conserver à l'esprit que la méthode d'authentification est différente au même titre que les permissions et la gestion des utilisateurs. Ainsi, il existe quelques restrictions notamment l'utilisation du "versioning" dans le bucket qui n'est au final pas compatible avec NFS.

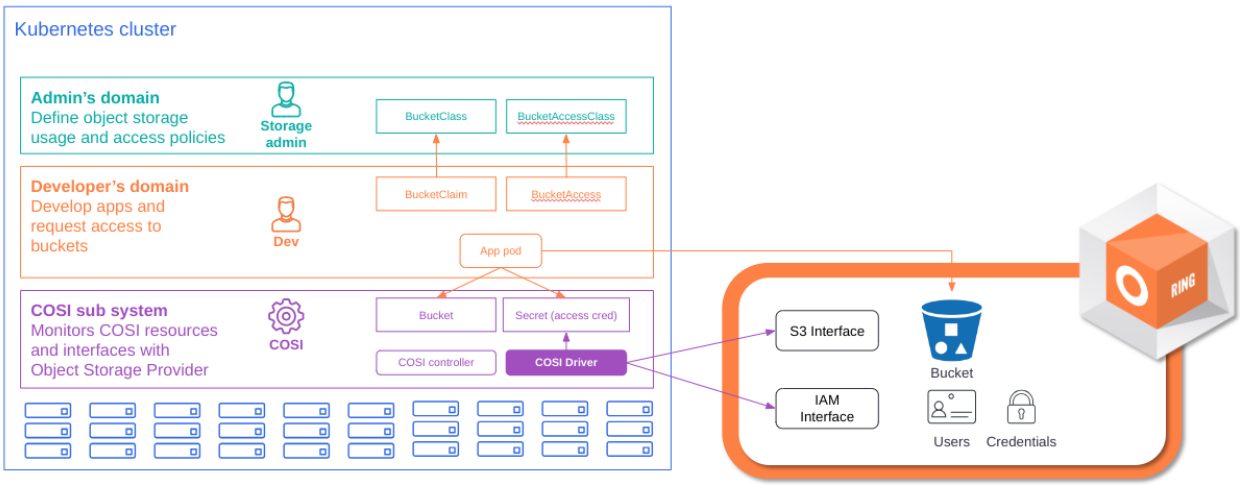


Toutefois, nous proposons chez nos clients désireux d'avoir un accès direct à un bucket la solution : mountpoint S3 de AWS permettant d'utiliser les méthodes d'authentification S3 et de monter un bucket sur un filesystem de type Linux par exemple. Une présentation de Mountpoint pour Amazon S3 est disponible ici :

https://aws.amazon.com/fr/about-aws/whats-new/2023/03/mountpoint-amazon-s3/

Driver COSI

La fonctionnalité COSI (**C**ontainer **O**bject Storage **I**nterface) est une spécification et une API introduites pour permettre l'intégration des solutions de stockage d'objets dans des orchestrateurs de conteneurs comme Kubernetes. Elle vise à fournir une approche standardisée pour consommer des services de stockage d'objets (telle que le RING), similaires à ce que le CSI (Container Storage Interface) offre pour le stockage en blocs et le stockage de fichiers.



Les fonctionnalités principales de COSI sont :

- Abstraction standardisée :
  COSI offre une interface unifiée pour gérer le stockage d'objets, quelle que soit la solution sous-jacente (RING de Scality, Amazon S3, Google Cloud Storage, etc.). Cela simplifie l'intégration des services de stockage d'objets avec des applications conteneurisées.
- Gestion de la vie des buckets :
  Avec COSI, les développeurs et les administrateurs peuvent provisionner, gérer et supprimer des buckets (conteneurs de stockage d'objets) de manière déclarative via des manifestes Kubernetes. Par exemple :
  - Création d'un bucket.

    - Gestion des permissions et des configurations associées.
    - Suppression du bucket lorsque son utilisation est terminée.
- Automatisation et contrôle via Kubernetes :
  COSI s'intègre à Kubernetes en utilisant des Custom Resource Definitions (CRDs), permettant de définir et de gérer des ressources telles que :
    - BucketRequest : une demande pour provisionner un bucket.
    - BucketClass : une classe définissant les politiques de provisionnement, comme le type de stockage ou la région.
    - Bucket : la ressource représentant le bucket réel.
- Séparation des responsabilités :
    - Les administrateurs définissent des stratégies et des configurations via des BucketClasses.
    - Les développeurs consomment des buckets via des BucketRequests, sans avoir besoin de connaître les détails techniques du backend.
- Extensibilité :
  Comme CSI, COSI est extensible et permet d'intégrer facilement différents fournisseurs de stockage d'objets en écrivant des pilotes spécifiques. Ces pilotes gèrent les interactions avec les API des fournisseurs.

Les cas d'usage sont variés mais on retrouve en général :

- Applications cloud-native :
  Les applications qui nécessitent un stockage d'objets pour stocker des fichiers, des journaux ou des données non structurées peuvent utiliser COSI pour automatiser la gestion des buckets.
- Portabilité :
  En utilisant une interface standardisée, les applications deviennent plus portables entre différents fournisseurs cloud ou solutions on-premises.
- Automatisation DevOps :
  Avec COSI, les pipelines DevOps peuvent inclure la création et la gestion de buckets comme partie intégrante du cycle de vie des applications.

Le processus de fonctionnement au sein de Scality RING est assez simple et peut se résumer au travers des actions ci-dessous :

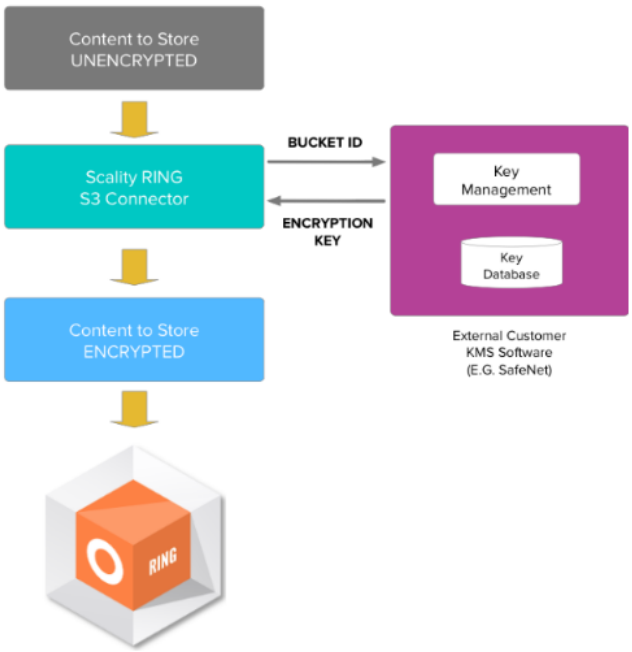
- Développeur : crée un objet BucketRequest pour demander un bucket.
- Kubernetes : utilise le driver COSI pour communiquer avec le backend du stockage d'objets.
- le RING : Provisionne le bucket demandé et retourne les informations nécessaires.
- Application : Consomme l’espace de stockage au travers du (ou des) bucket(s) via l'URL et les informations d'accès fournies (AK/SK).

En résumé, la fonctionnalité COSI de RING simplifie, standardise et automatise l'utilisation du stockage d'objets pour les applications conteneurisées tout en offrant une portabilité et une extensibilité pour vos différents environnements.

Chiffrement

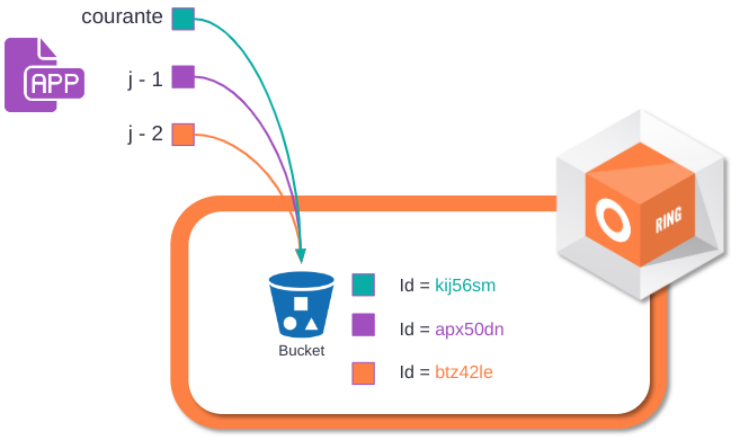
Vos données sont sécurisées au travers d'un chiffrement qui peut s'opérer de bout en bout ; c'est à dire depuis vos applications et jusqu'aux disques dans le RING Scality.

Durant le transport des données (entre l'application et le stockage objet), les données sont chiffrées via le protocole TLS/SSL sur la session HTTPS.
Pour les données stockées et restant dans le bucket, il est possible d'activer le chiffrement SSE via la méthode S3 Bucket Encryption. SSE est pris en charge pour le stockage de données chiffrées @REST, via des algorithmes de chiffrement sécurisés AES256 (OpenSSL library). Une clé unique de chiffrement est générée pour chaque objet qui est elle-même chiffrée par une clé de type "**Master Key**".
Cette clé "Master Key" peut être locale à Scality RING ou bien gérée par un KMS externe (Key Management Services) au travers du connecteur S3. Le protocole utilisé dans ce cas est **KMIP** en version 1.2 et supérieur.



Versioning

Le versioning fait partie des fonctionnalités disponibles au niveau de chaque bucket S3. Il est désactivé par défaut et peut être activé à la demande puis suspendu si nécessaire.
Dans le cas où l'application en amont ne sait pas gérer le versioning des objets, il est possible de créer des règles ILM (LifeCycle Rules) permettant de supprimer automatiquement les versions courantes ou ultérieures en fonction de paramétrage défini (ex : date). Plusieurs critères et options sont possibles sur les règles ILM. La documentation RING les expose.
Il n'existe pas de nombre de versions maximum dans le protocole S3, ni dans RING. Une nouvelle version d'un objet et un objet en lui-même et dispose des mêmes attributs.
Au travers des API S3, la mise en place d'une suppression sur un nombre de versions est envisageable pour un objet.



Immuabilité des données

Le RING Scality prend en charge S3 Object Lock. Cette fonctionnalité de l'API AWS-S3 permet de faire du WORM (write once read many) pour les objets S3 :

- Empêche un objet d'être supprimé ou écrasé pendant une durée fixe ou indéfinie
- Les suppressions sont bloquées et les écrasements sont gérés via la gestion des versions
- Activé au niveau du bucket avec une granularité au niveau de l'objet

Deux niveaux de contrôles :

- Période de rétention : pour empêcher les suppressions ou les écrasements pour une durée prédéfinie
- Retenue légale : pour empêcher la suppression ou l'écrasement jusqu'à ce que la retenue soit supprimée

Deux modes de rétention :

- Mode de gouvernance : le verrouillage peut être contourné par l'utilisateur root du compte, privilèges supprimés possibles. Destiné à la protection des données contre la compromission du compte et les acteurs malveillants
- Mode de conformité (WORM Legal): le verrouillage ne peut pas être contourné même par l'utilisateur root du compte, aucune suppression possible. Destiné à la conformité aux réglementations telles que SEC 17a-4, CFTC et FINRA

Scality a reçu la certification de conformité SEC 17a-4 par Cohasset Associates :

Multi-tenancy

Les utilisateurs S3 disposent d'un accès à leurs buckets via une authentification S3 par jeux de clés : access key et secret key. Il s'agit d'un couple : identifiant d'une quinzaine de caractères alphanumériques et credentials d'une cinquantaine de caractères alphanumériques.
Un utilisateur s'intègre dans un groupe d'utilisateurs et peut posséder plusieurs clés S3 permettant de faire une rotation.
Chaque compte S3 (ou S3 root) peut également gérer des comptes IAM (Identity Access Management) permettant d'être granulaires sur les méthodes autorisées ou non par utilisateur IAM. Chaque groupe ou utilisateur IAM se voit attribuer une politique lui donnant des droits et des ressources auxquelles il a accès (allow or deny policy).
L'interface graphique offre un client S3 web permettant aux utilisateurs de naviguer dans leurs espaces de travail/buckets.

Le processus de création de buckets s'effectue simplement au niveau des utilisateurs S3 ou bien des utilisateurs IAM. Le bucket en lui-même dispose de permissions par défaut (private = limitant les accès indésirables). Il est possible de modifier le paramétrage des buckets après l'initialisation/création : versioning, chiffrement, etc.

Les ressources sont rendues accessibles aux utilisateurs IAM au travers de politiques IAM (format JSON standard).
A noter qu'il n'est bien évidemment pas possible de passer un bucket existant (non WORM dès l'origine) en mode WORM par la suite.

Voici les caractéristiques de notre implémentation d'IAM au sein de RING :

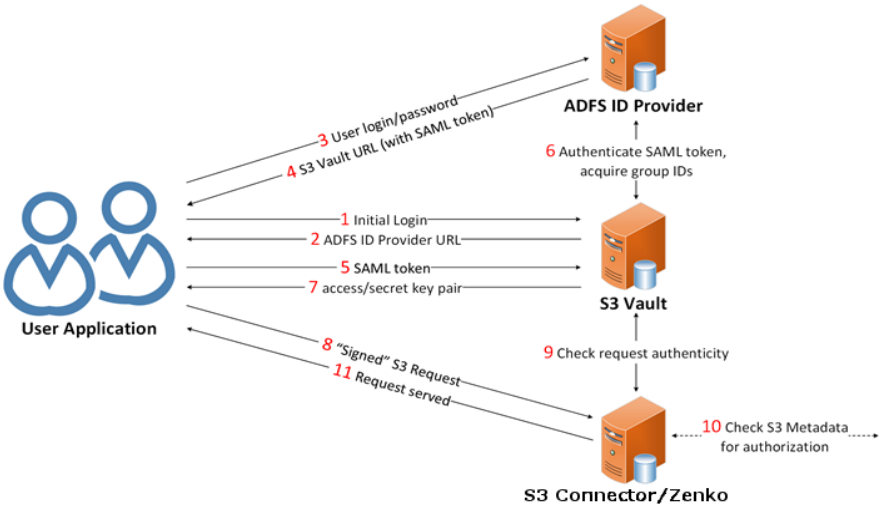
S3 API Capabilities | IAM Capabilities

IAM Description	Amazon Capabilities	S3 Connector Capabilities
Number of users per account	5000	5000+
Number of groups per account	100	100+
Number of roles in an AWS account	250	250
Number of access keys per user	2	2+
Number of groups a user can be associated with	10	10+
Number of managed policies per account	1000	1000+
Number of managed policies attached to an IAM user, group, or role	10	10+
Quota per account	no	yes

Plus finement, il y a trois modes complémentaires pour gérer les accès au Stockage S3. Plusieurs options existent pour obtenir et gérer les clefs en fonction des besoins :

- Le premier mode repose directement sur l'interface IAM (Identity and Access Management) du RING. Un compte provisionné dans le Vault IAM fait une demande, soit par API, soit par l'interface web des clefs. Les clefs ne sont communiquées qu'une seule fois (si elles sont perdues, il faut les remplacer). Les clefs peuvent avoir une durée de vie paramétrable ou indéterminée, et peuvent être activées ou désactivées à tout moment. Pour les tiers accédant régulièrement à la plateforme, une politique d'accès précise peut être appliquée sur les actions, les buckets et avec un nombre limité d'adresses IP, par exemple, ce qui permet un contrôle complet de l'accès. Les clefs peuvent être désactivées et réactivées pour bloquer temporairement l'accès. Cette approche peut aussi être utilisée pour les administrateurs, pour des utilisateurs spécifiques qui ont besoin d'accès privilégiés, ou pour des outils de maintenance qui travaillent de façon permanente.
- Le mode le plus pertinent pour la majeure partie des utilisateurs est basé sur la notion d'identité fédérée, et peut être basé sur l'infrastructure des utilisateurs à travers l'interface SAML 2.0 de solutions comme F5 ou ADFS, etc.

Le diagramme ci-contre montre les étapes de ce processus.



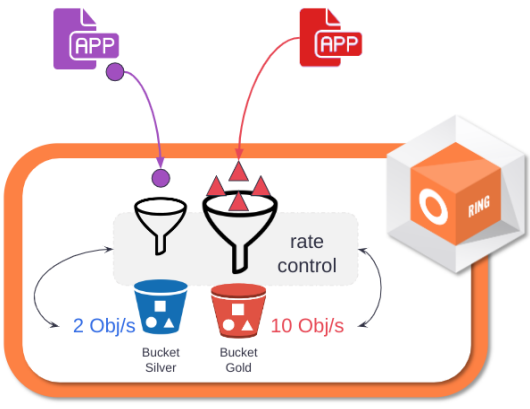
L'ADFS peut gérer de multiples annuaires et le Vault Scality peut gérer de multiples IP providers SAML. L'utilisateur s'identifie à travers le SSO SAML et obtient un jeton qu'il peut utiliser pour faire une requête des clefs. Selon son identité et les groupes auxquels elle appartient, la personne aura le droit d'assumer des rôles définis dans l'IAM. L'utilisateur ou l'application prend alors les clefs pour interagir avec l'interface S3 de façon classique. Dans cette approche, la validité des clefs est assez courte, le temps de faire des opérations immédiates. Typiquement les appels scriptés font la demande de renouvellement de clefs à chaque lancement. Alternativement, les clefs peuvent être également demandées pour des périodes plus longues, par exemple quotidiennement, pour permettre à un utilisateur dans une session Linux Kerberos de passer des commandes.

- La dernière approche, surtout utile pour un usage ponctuel, est décrite dans les étapes 1 et 2 du diagramme. Une requête de type login GUI web est alors dirigée vers le Vault Scality, puis redirigée soit vers le SSO, soit directement vers l'identity provider. L'utilisateur s'authentifie avec l'interface web, éventuellement en MFA, et reçoit ensuite les clefs à travers l'UI web. La création d'une page web pour faire la requête est facilement implémentable. L'utilisateur qui obtient les clefs peut ensuite les insérer dans une application type Cyberduck ou autre, en faisant directement des requêtes S3 standards. C'est à l'utilisateur de refaire la manipulation quand les clefs sont périmées.

La première approche est la mieux adaptée pour les configurations statiques : enregistrement des clés dans une application, alors que les deux autres permettent au RING d'être intégré dans l'environnement SSO et de ne pas multiplier les comptes ni la gestion des mots de passe.

QoS

Plusieurs workloads peuvent être dirigés sur un cluster RING. Afin de les séparer, il est possible de créer des Endpoints S3 différents ou de les isoler sur des réseaux différents. En cas de besoin de performance spécifique pour un workload par rapport à un autre, il est recommandé de dédier des connecteurs S3 y compris pour le multi-tenant.



Le RING 9.5 implémentera une fonctionnalité de "rate control" dont le but est de fournir une réponse sur la partie Front-End au DDoS (limitation des requêtes) mais aussi d'éviter qu' un bucket ne soit trop gourmand en termes de requêtes par rapport aux autres.

Il sera ainsi possible de mettre en place une priorisation de bucket via le contrôle du nombre de requête et ainsi limiter les effets de bords.

Il n'y a pas de QoS possible à ce jour sur la bande passante IN / OUT.

Quota

Les quotas peuvent être gérés pour les différents protocoles : NFS, SMB et S3.
Pour les connecteurs Fichier : les quotas sont appliqués par l'administrateur de stockage via le superviseur et peuvent être définis en fonction de la capacité ou d'un nombre de fichiers. Les quotas peuvent être définis au niveau du volume, de l'utilisateur ou du groupe. Lors de l'affichage des propriétés du partage réseau, les utilisateurs verront la quantité de stockage allouée dans les limites de leur quota. Il existe une limite souple et une limite stricte - une fois que la limite stricte est atteinte, l'utilisateur recevra un message l'informant qu'il ne peut plus écrire. Les administrateurs peuvent consulter les rapports quotidiens pour voir si des utilisateurs sont signalés dépassant leur limite souple ou stricte. Une période de grâce est également configurable si vous le souhaitez.
Nous avons développé un CLI (*squotactl*) qui peut être appelé par un outil d'automatisation pour la remontée d'information, notamment le reporting.

Pour les connecteurs S3, Scality supporte les quotas au niveau du Account. Scality fournit une API UTAPI (Service Utilization API) pour le suivi de l'utilisation des ressources et le reporting des mesures. UTAPI inclut des informations sur la capacité de stockage de Scality RING, le nombre d'octets transférés dans le service et le nombre d'opérations effectuées sur le service. Il étend l'API de base AWS S3 REST, permettant une solution complète de suivi de l'utilisation des ressources sur site, requise par les fournisseurs de services pour les fonctionnalités de génération de rapports externes ou de facturation interne.
UTAPI fournit une fonctionnalité intégrée de création de rapports sur les ressources pour le connecteur Scality S3, capable de suivre et de signaler les mesures d'utilisation suivantes via une API RESTful:

- Capacité de stockage en octets
- Nombre d'objets
- Utilisation du réseau par unité de temps
- Nombre d'opérations par unité de temps

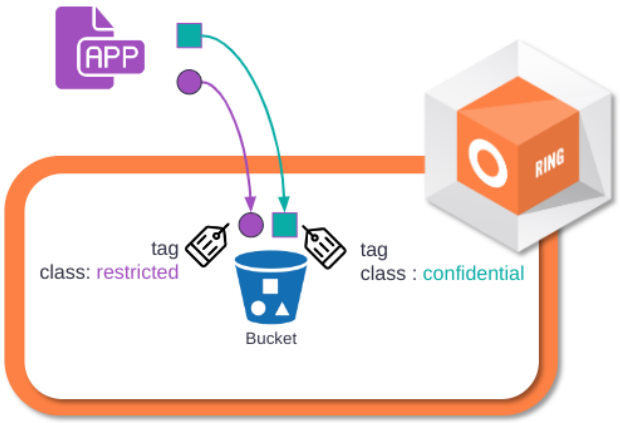
UTAPI suit toutes ces métriques et les rend disponibles pour la génération de rapports à quatre niveaux de granularité, afin de prendre en charge un large éventail d'exigences en matière de rapports d'entreprise:

- Global (sur l'ensemble du service)
- Niveau de compte
- Niveau de l'utilisateur
- Niveau du bucket

TAG

Les tags dans le protocole S3 offrent une méthode simple et puissante pour ajouter des métadonnées personnalisées aux buckets ou aux objets stockés. Ces tags sont constitués d'un couple "clé : valeur" permettant, par exemple, de différencier des données d'un environnement par rapport à un autre.

- clé : valeur → Environnement : Production ; Environnement : Qualité
- clé : valeur → Project : Alpha ; Project : RC



Voici quelques autres cas d'utilisation des tags par nos clients :

- Organisation et classification

Les tags permettent de structurer et de catégoriser les buckets et objets de manière logique. Cela facilite la gestion des données, surtout dans les environnements où de nombreux objets ou buckets coexistent.
Exemple : Taguer des objets en fonction du projet (Project: Revamp) ou de leur sensibilité (Confidential: Yes).

- Suivi des coûts et des allocations budgétaires

Les tags sont très utiles pour attribuer les coûts liés au stockage et à l'utilisation des ressources S3 à des équipes, départements ou projets spécifiques. Ainsi il est possible de détailler la répartition des dépenses.
Exemple : Utiliser un tag Department: Finance pour analyser le coût des buckets utilisés par l'équipe finance.

- Gestion des permissions (IAM Policies)

Les tags peuvent être utilisés pour définir des politiques IAM (Identity and Access Management) basées sur des conditions. Cela permet un contrôle fin de l'accès aux ressources S3.
Exemple : Restreindre l'accès à tous les objets avec le tag Confidential: Yes à une équipe spécifique.

- Cycle de vie des objets

Les règles de gestion du cycle de vie dans S3 (S3 Lifecycle) peuvent être configurées en fonction des tags. Cela permet d'automatiser le transfert des objets entre les classes de stockage ou leur suppression.
Exemple : Les objets avec le tag Temporary: True peuvent être supprimés après 30 jours.

- Recherche et filtrage

Dans des systèmes contenant des millions d'objets, les tags offrent un moyen rapide de rechercher ou de filtrer des ressources. Les services AWS comme S3 Inventory ou Amazon Macie peuvent également tirer parti des tags pour analyser ou sécuriser les données.

- Sécurité et conformité

Les tags permettent d'identifier rapidement les objets soumis à des régulations spécifiques (par exemple, GDPR, HIPAA) et de s'assurer qu'ils respectent les politiques de conformité définies.
Exemple : Taguer les données sensibles avec Compliance: GDPR pour appliquer des restrictions supplémentaires.

Gestion du cycle de vie (ILM)

Les règles d'ILM (expiration lifecycle rules) sont disponibles et configurables directement depuis le navigateur S3 web de RING et portent sur la version courante, précédente, les DMs, les MPU avec des filtres optionnels tels que : prefix et tags.

Le module eXtended Data Management (XDM) contient un ensemble de fonctionnalités qui étend la gestion des données RING sur plusieurs RINGs, NAS et nuages publics.
Ceci est conçu pour prendre en charge les nouveaux cas d'utilisation de la gestion des données hybrides-cloud, notamment :

ARCHIVE RING à cloud :

- Décharger les données RING anciennes/inutilisées vers un niveau cloud bon marché pour les archives à long terme
- Libérer la capacité RING tout en conservant des données à des fins de compiance ou de réglementation
- Copie des données RING (en tout ou en partie) dans le cloud pour la récupération après sinistre

Les fonctionnalités cloud avancées de XDM incluent :

- Recherche de métadonnées
- Transition automatisée des données
- Expiration des données - cycle de vie
- Réplication inter région (CRR) dans le Cloud

Afin d'activer les fonctionnalités XDM, il est nécessaire que des serveurs supplémentaires effectuent les flux de travail et gèrent la base de données de métadonnées. Il est possible d'installer XDM sur un seul petit serveur 1U , mais il est conseillé d'utiliser 3 nœuds afin d'avoir de la haute résilience et le quorum de métadonnées.

Durabilité de vos données

La durabilité des données sur une plateforme de stockage objet est liée à plusieurs principes notamment au niveau de l'architecture et des moyens mis en place pour palier à une panne éventuelle. A titre d'exemple, le service de stockage Amazon S3 propose une durabilité de 99.999999999% ou "11 nines" (https://docs.aws.amazon.com/whitepapers/latest/real-time-communication-on-aws/data-durability-and-ha-with-persistent-storage.html).
Plusieurs fonctionnalités et mécanismes sont disponibles sur notre technologie RING afin de garantir une protection élevée des données de nos clients.
Pour mémo, il n'y a aucun RAID utilisé pour protéger les objets dans le cluster Scality mais des algorithmes à plusieurs parités +2, +3, +4, +5, +7 ... permettant de durabilité sans faille.

- **Redondance et protection des données** : Trois mécanismes sont disponibles pour protéger les données localement et géographiquement : facteur de réplication (RF ou COS - Class Of Service), Erasure Coding (Scality ARC pour Advanced Resilience Configuration), la géo répartition permettant de couvrir de multiples sites.
- **Autoréparation** : Le RING Scality offre un mécanisme complet et entièrement automatisé qui permet de détecter sans intervention manuelle les défaillances au niveau de l'objet ou du nœud et des objets reconstruits. Cela protège de la corruption silencieuse de données suite à des problèmes ou erreurs sur les disques. Cette fonctionnalité est entièrement configurable, permettant à l'administrateur d'accélérer la resynchronisation des données ou de donner la priorité à une application particulière. Cette fonctionnalité sert à satisfaire les demandes de disponibilité les plus strictes.

- **Auto-équilibrage** : En réponse à des changements de configuration ou de topologie, comme la perte ou ajout de nœuds, le RING rééquilibre automatiquement l'espace des clefs à travers les nœuds de stockage. Cette fonction est configurable par la console d'administration, permettant à l'administrateur d'adapter les paramètres spécifiques destinées à améliorer ou à limiter la fonction d'équilibrage.
- **Répartition des données** : Le système garantit la répartition des données en tout temps. Les dossiers et fichiers sont toujours distribués à travers tous les serveurs de stockage pour garantir la performance, la durabilité et la disponibilité en cas de panne de disque ou serveur.
- **Elasticité** : Le RING est élastique et sa capacité peut grandir ou diminuer en toute transparence. Le RING a été développé pour être totalement autonome et réagir aux variations de charge dues à des pannes réseau ou de disque, des ajouts et des ajustements de configuration matérielle.
- **Redondance d'accès** : De multiples connecteurs « stateless » sont configurés pour maintenir l'accès à l'infrastructure de données. L'accès parallèle aux nœuds de stockage et les copies multiples des données permettent un service cohérent et continu aux applications. L'architecture standard doit inclure un (ou des) load-balancer en frontal des connecteurs S3 permettant d'assurer l'envoi des requêtes sur ces derniers.

Redundancy	Overhead	Nb of system that can fail	Durability
RAID 5	X 1.2	1	98.5 % (2 nines)
RAID 6	X 1.3	2	99.9998 (6 nines)
ARC(4,2)	X 1.5	2	99.99999% (7 nines)
Replication X3	X 3	2	99.9999999% (9 nines)
ARC (9,3)	X 1.33	3	99.9999999% (9 nines)
Replication X4	X 4	3	99.999999999% (11 nines)
ARC (12,4)	1.3	4	99.999999999% (11 nines)
ARC (8,4)	1.5	4	99.9999999999% (12 nines)
Replication X5	X5	5	99.999999999999% (14 nines)
ARC(7,5) on 3 sites	X1.7	1 site and 1 server	99.999999999999% (14 nines)
ARC(5,7) on 2 sites	X2.4	1 site and 1 server	99.999999999999% (14 nines)

Pour les environnements de Production, Scality recommande d'opter pour une redondance comprise entre "**9 nines**" et "**14 nines**".

COS et ARC

De par notre technologie et nos optimisations, les petits objets (<60 KB) seront protégés par un mécanisme **COS (facteur de réplication)** et les objets plus grands (>60 KB) par un mécanisme **ARC (Advanced Resilience Configuration)**.
Nous allons ainsi permettre une protection optimum dans un même bucket contenant des objets de

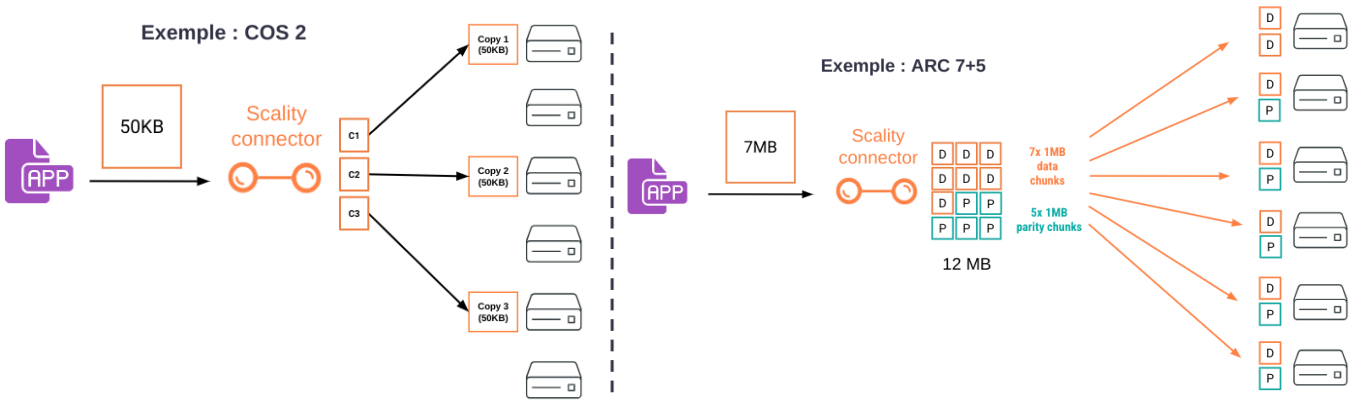
taille différente.

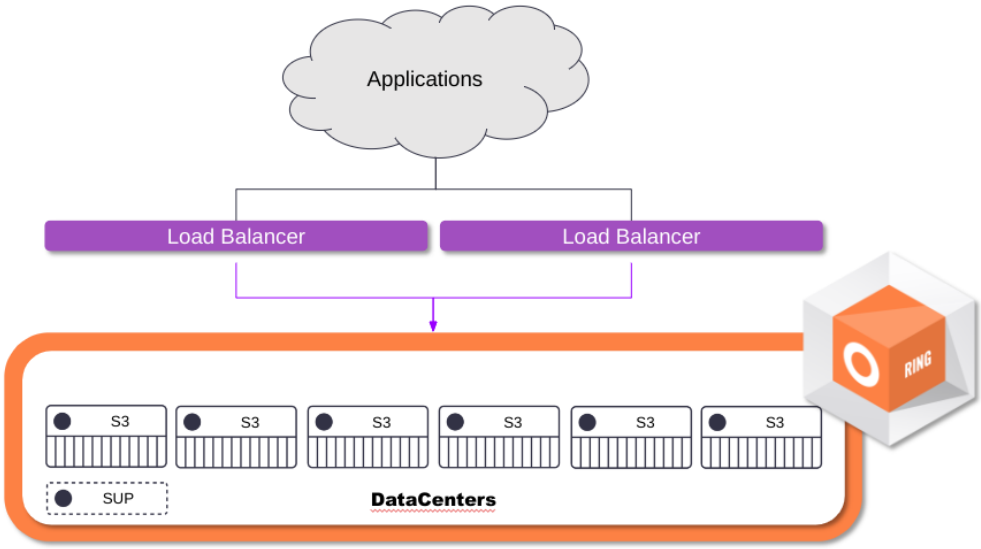
L'ARC de Scality implémente les techniques de codage Reed-Solomon, pour stocker de grands objets avec un ensemble étendu de « morceaux » ou de parité, au lieu de plusieurs copies de l'objet d'origine. L'idée de base est de diviser un objet en plusieurs morceaux (k), et d'appliquer un codage mathématique pour produire un ensemble supplémentaire de morceaux de parité (m). L'ensemble de morceaux résultant (k+m) est ensuite distribué sur les nœuds RING, ce qui permet d'accéder à l'objet d'origine tant que n'importe quel sous-ensemble de données ou de morceaux de parité sont disponibles (minimum k). Cela fournit un moyen de stocker un objet avec une protection contre les défaillances de m composants, avec seulement k/(k+m) d'espace de stockage supplémentaire nécessaire (le ratio brut vs. utile).
Les coefficients COS et ARC diffèrent par rapport au nombre de nœuds et à l'architecture du cluster : 1 site, 2 sites stretched, 3 sites géo.
COS 2 = 1 original + 2 copies = ratio raw/usable de 33%
COS 3 = 1 original + 3 copies = ratio raw/usable de 25%
ARC 7+5 = 7 fragments + 5 parités = ratio raw/usable 58%
ARC 5+7 = 5 fragments + 7 parités = ratio raw/usable 42%

La capacité du stockage, la protection et la performance sont donc améliorées en choisissant de manière automatique le meilleur mécanisme de protection.

Load-balancers

Pour le protocole S3, la gestion du load-balancing est indispensable dans les architectures RING. Elle doit être mise en place au moyen de load-balancer externes dirigeant le flux S3 sur les connecteurs RING. Le minimum requis est la gestion de flux et les healthcheck Layer 4 (TCP/IP). Toutefois, il est recommandé d'opter pour des healthcheck Layer 7 disponibles sur les load-balancers communs : F5, HAProxy, NGINX, Kemp, etc. Nous pouvons fournir lors du déploiement les "best practices" à ce sujet afin d'optimiser la Production.
Afin d'éviter les SPOF, il est conseillé d'avoir au moins 2 load-balancers (actif/passif ou actif/actif) bare-metal ou sous forme de machine virtuelle.





Scality peut vous accompagner sur la mise en place de load-balancer sous forme de machines virtuelles intégrant les solutions : HA Proxy & KeepAlived.
Les ressources nécessaires sont liées aux différents cas d'usage et au nombre de sessions qu'il faut gérer. L'avantage des machines virtuelles réside bien évidemment dans la souplesse de pouvoir augmenter ses ressources assez rapidement et assez facilement.
En standard, il est recommandé :

- 4 vCPU minimum
- 16 Go de mémoire minimum
- 1 à 2 ports réseau agrégés → 10Gb et plus (throughput global et attendu). Prendre en compte les entrées/sorties sur des réseaux/VLANs différents (segmentation).
- 50 Go de disque minimum (et davantage si il y a une conservation des logs)

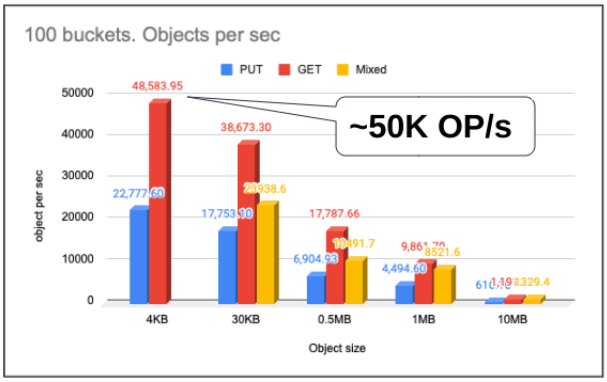
Linéarité des performances

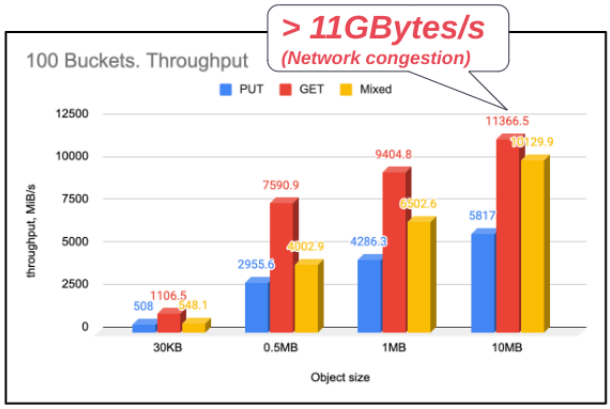
Le cluster RING permet d'évoluer comme vu précédemment de différentes façons (scale-IN & OUT). L'ajout de nœud dans le cluster permet d'évoluer de façon linéaire sur les performances notamment en terme de bande passante (RING de type HDD). Le tableau ci-dessous indique les performances sur une version 8 de RING avec des châssis de 24 slots.

Number of Fully Provisoned Servers (24x HDD)	Useable TB (12TB HDD)	Throughput Maximum Values (MBps)			
		S3 Write	S3 Read	File Write	File Read
Single Site					
3	472	2086	3202	2039	1706
6	1063	4255	5540	3246	3627
2 Sites Stretched					
6	696	2659	3462	2029	2267
8	929	2947	3837	2249	2513
12	1393	4432	5771	3381	3779
3 Sites Stretched					
6	976	3723	4847	2840	3174
9	1464	4653	6059	3550	3967
12	1942	6205	8079	4734	5290

Étant agnostique au matériel, le RING supporte également les environnements "full flash" fournissant davantage de performances d'un point de vue : OP/s (opérations par seconde) et bande passante. Ci-

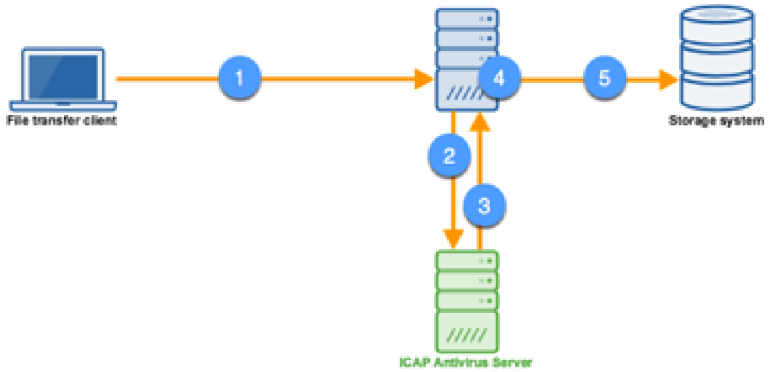
dessous quelques exemples d'un système "full flash" de 6 nœuds.





Compatibilité Antivirus

L'objectif est l'analyse efficace des fichiers et objets générés par plusieurs applications puis stockés et mis à jour dans le Scality RING en fournissant des fonctionnalités équivalentes à ICAP : l'analyse antivirus ICAP, en particulier, permet de déporter toutes les tâches d'analyse antivirus du serveur de stockage vers un serveur ICAP AV. Ci-dessous un schéma d'architecture standard d'une solution ICAP :

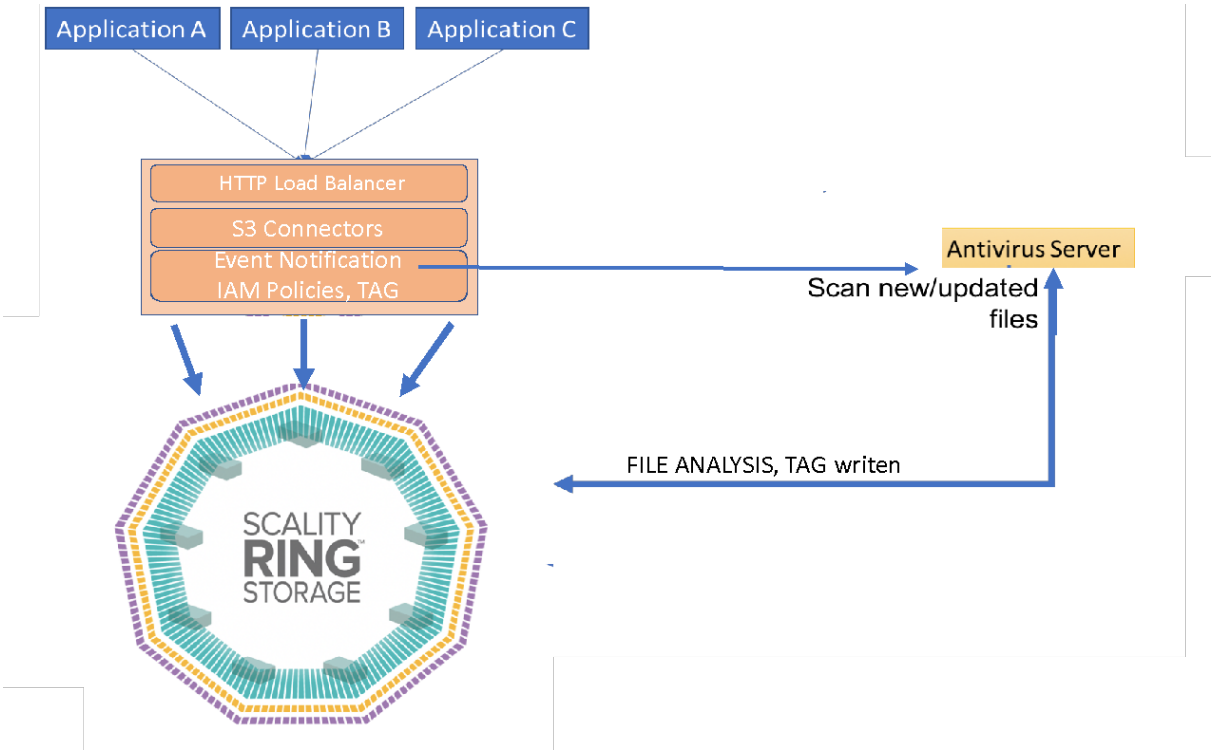


Scan d'objets écrits via le connecteur S3

Lors de l'utilisation du connecteur S3, les anti-virus peuvent être utilisés "outband" lors de l'acquisition, en effectuant l'analyse en parallèle que les objets soient transférés dans les buckets du RING. Ainsi on évite tout goulot d'étranglement. La donnée est analysée presque en temps réel, au fur et à mesure que celle-ci est écrite sur le RING.

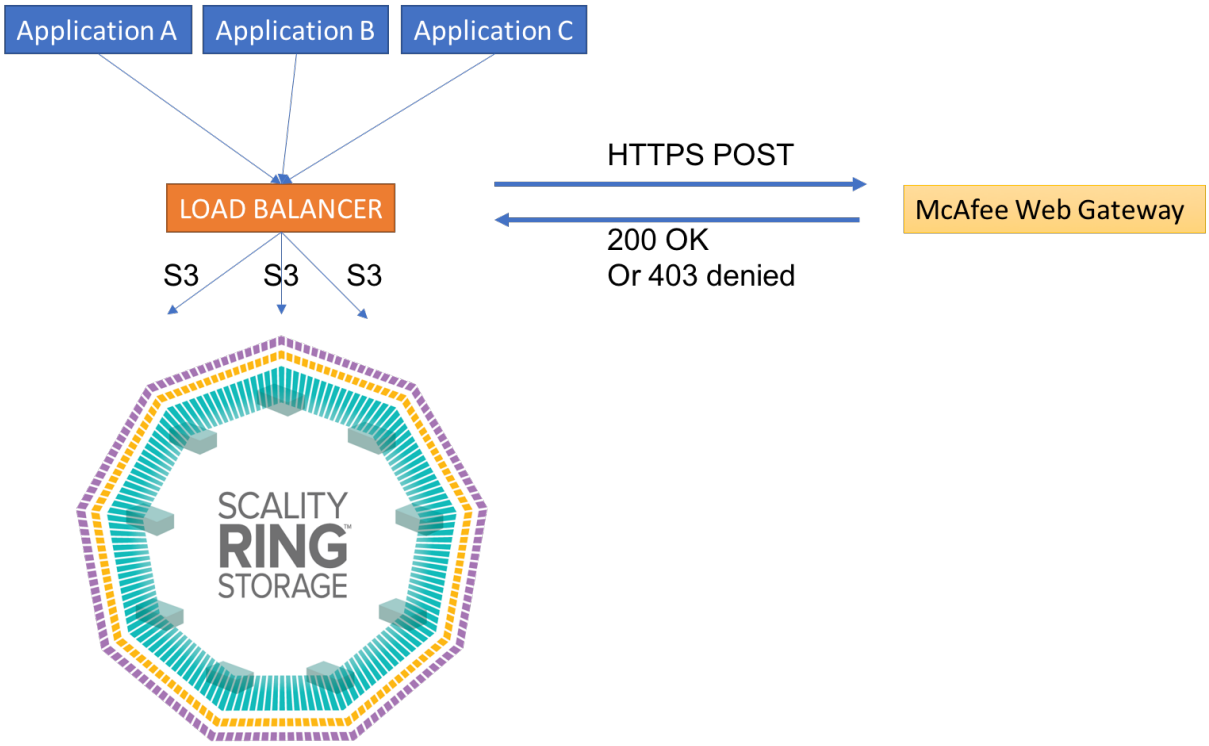
Pour ce faire, les antivirus tirent parti de la technologie « S3 Event Notification » pour analyser uniquement les objets récemment uploadés ou modifiés sur le RING. C'est le cas notamment de l'antivirus open-source Clamavd ou de Symantec (https://support.symantec.com/us/en/article.tech249190.html) qui peuvent scanner tous les objets pour lesquels il ont reçu une notification. Il est par ailleurs possible d'utiliser les TAG et les IAM Policies pour plus d'optimisation et ainsi bénéficier de fonctionnalités similaires à ICAP. La fonctionnalité TAG permet à un antivirus de marquer les objets comme «CLEAN» ou «INFECTED». Les IAM Policies peuvent être configurées pour empêcher quiconque de lire un objet dont le TAG est « INFECTED » , ou différent de « CLEAN ».

Exemple de fonctionnement avec un anti-virus analysant les données pour lesquelles il a reçu une notification :



Par ailleurs, les anti-virus peuvent faire des requêtes « list-bucket » pour lister les objets déjà présents sur le RING et ainsi les re-scanner. Un entête rescan-date peut être rajouté sur les objets.

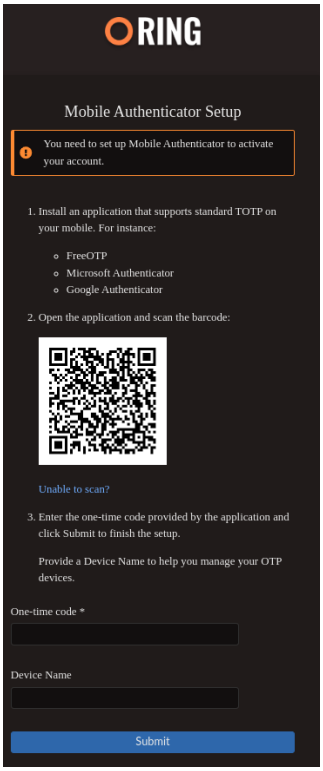
Pour finir, les anti-virus externes peuvent être utilisés "inband" lors de l'acquisition, en effectuant l'analyse avant que les objets ne soient transférés dans les buckets du RING. C'est par exemple le cas de l'antivirus McAfee et sa Web Gateway utilisée en mode reverse proxy. Cette configuration permet d'analyser le contenu de la donnée avant de l'envoyer sur le RING. Dans le cas où le contenu est infecté, McAfee Web Gateway renvoie une réponse 403 « Denied » à l'application.
McAfee Web Gateway intercepte le contenu avant qu'il n'atteigne les connecteurs S3, le traite, puis le bloque ou le transmet, en fonction des résultats de l'analyse. Si le contenu est bloqué, il n'atteint jamais les connecteurs S3, et donc n'atteint jamais le RING.
Exemple d'intégration des connecteurs S3 avec un anti-virus McAfee Web Gateway bloquant l'écriture de la donnée s'il y a un virus.



Plus d'information sur l'anti-virus McAfee et sa fonctionnalité Web Gateway : http://www.mcafee.com/de/resources/data-sheets/ds-web-gateway-reverse-proxy.pdf

Interface d’administration et supervision

Pour gérer et surveiller le RING, Scality fournit un ensemble complet d’outils, avec diverses interfaces. Celles-ci incluent une interface graphique Web (le superviseur), une interface de ligne de commande qui peut être scriptée (RingSH) et, pour une utilisation avec les consoles de surveillance SNMP standard, le RING fournit des MIB et des traps conformes à SNMP.



L’authentification se produit au travers d’un couple login/mot de passe et d’un complément MFA.

Supervisor Web Management GUI

Le superviseur est l’interface graphique de gestion du RING en format web. Il fournit une surveillance et une gestion visuelle du logiciel RING, ainsi que de la couche de plate-forme physique sous-jacente.

Le superviseur prend en charge de manière native le contrôle d'accès basé sur le rôle (RBAC), afin de permettre à plusieurs rôles administratifs dotés d'autorisations différenciées de gérer et de surveiller le RING via l'API et les outils.

Le superviseur fournit une page principale de tableau de bord contenant des vues graphiques RING, notamment les serveurs, les zones et les nœuds de stockage composant le RING, ainsi que des fonctionnalités de navigation permettant de rechercher des informations détaillées sur chaque composant, ainsi que des pages dédiées aux opérations, à la gestion et à la fourniture des services RING. Le superviseur fournit également des statistiques sur les performances, la consommation de ressources et des mesures de l’état de santé par le biais d’un riche ensemble de graphiques.

Un code couleur vous prévient automatiquement de la perte d’un disque et/ou châssis (en plus de remontées d’alertes SNMP, nRPE ou autres alertes configurées).

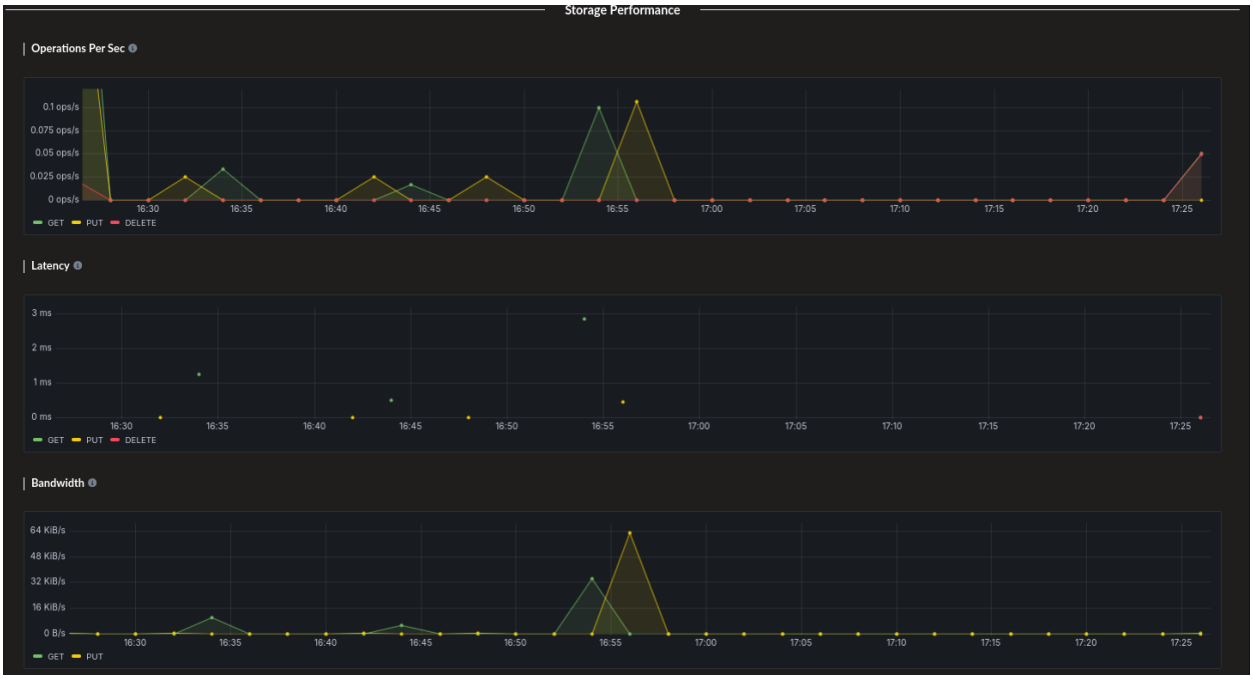
Le tableau de bord principal du superviseur est le suivant :

Chaque composant du RING détient un “exporter” permettant de collecter des informations sur la

nature du composant, les capacités attenantes mais aussi des métriques :

Name	Status	Zone	Hardware	Capacity	Roles
dc1-cifs-0	●	dc1	VM	0 B	SOFS/SMB
dc1-localfs-0	●	dc1	VM	0 B	SOFS/LocalFS
dc1-nfs-0	●	dc1	VM	0 B	SOFS/NFS
supervisor	●	dc1	VM	0 B	Supervisor
dc2-cifs-0	●	dc2	VM	0 B	SOFS/SMB
dc2-localfs-0	●	dc2	VM	0 B	SOFS/LocalFS
dc2-nfs-0	●	dc2	VM	0 B	SOFS/NFS
dc1-store-1	●	dc1	VM	161.06 GB	S3, S3 MD, Simple REST, Storage
dc1-store-2	●	dc1	VM	161.06 GB	S3, S3 MD, Simple REST, Storage
dc2-store-1	●	dc2	VM	161.06 GB	S3, S3 MD, Simple REST, Storage
dc2-store-2	●	dc2	VM	161.06 GB	S3, S3 MD, Simple REST, Storage
dc3-store-1	●	dc3	VM	161.06 GB	S3, S3 MD, Simple REST, Storage
dc3-store-2	●	dc3	VM	161.06 GB	S3, Simple REST, Storage

En cliquant sur un composant, on descend dans les informations afin d'obtenir des métriques de performances par exemple :



RingSH Command Line Interface

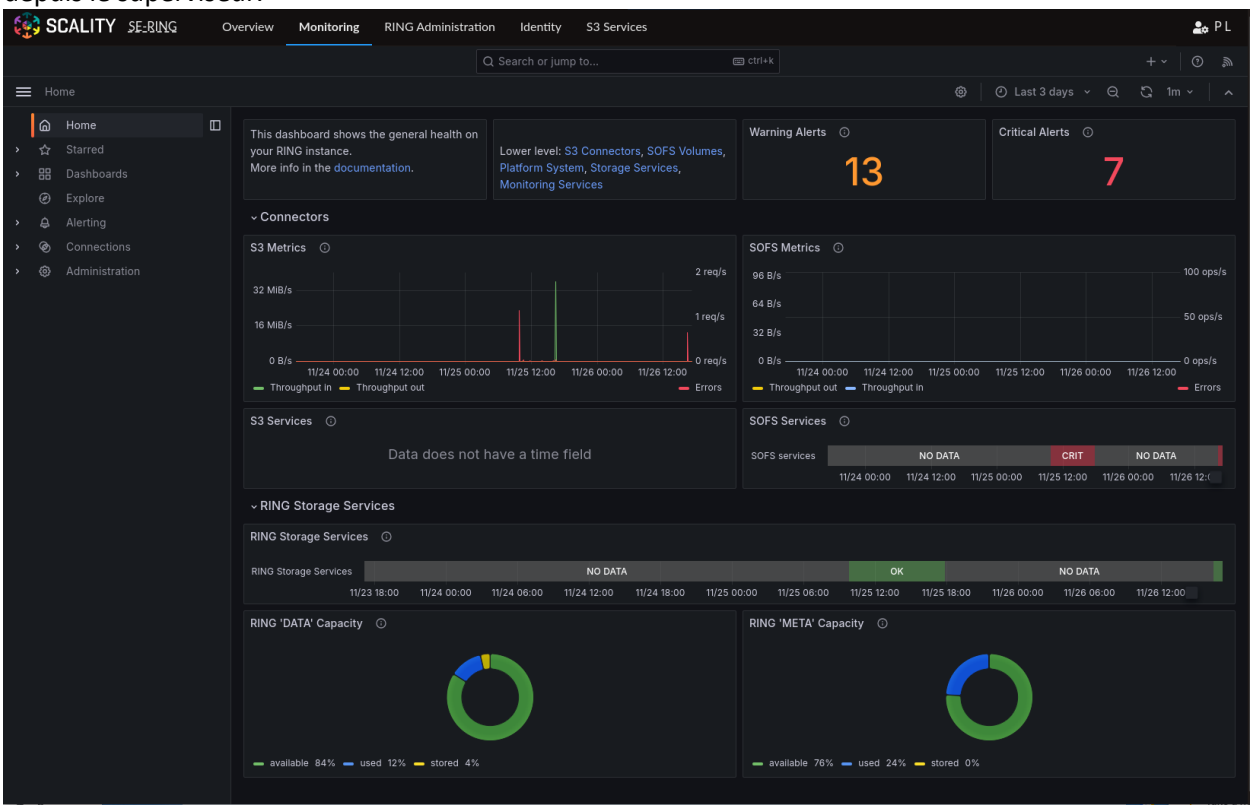
RingSH est une interface de ligne de commande scriptable pour la gestion et la surveillance du RING, qui peut être utilisée sur l'hôte Supervisor ou sur tout serveur de stockage pour gérer les composants RING. RingSH fournit un ensemble complet de commandes permettant de gérer l'ensemble du système, ainsi que d'accéder aux statistiques et aux métriques de santé.

SNMP Monitoring

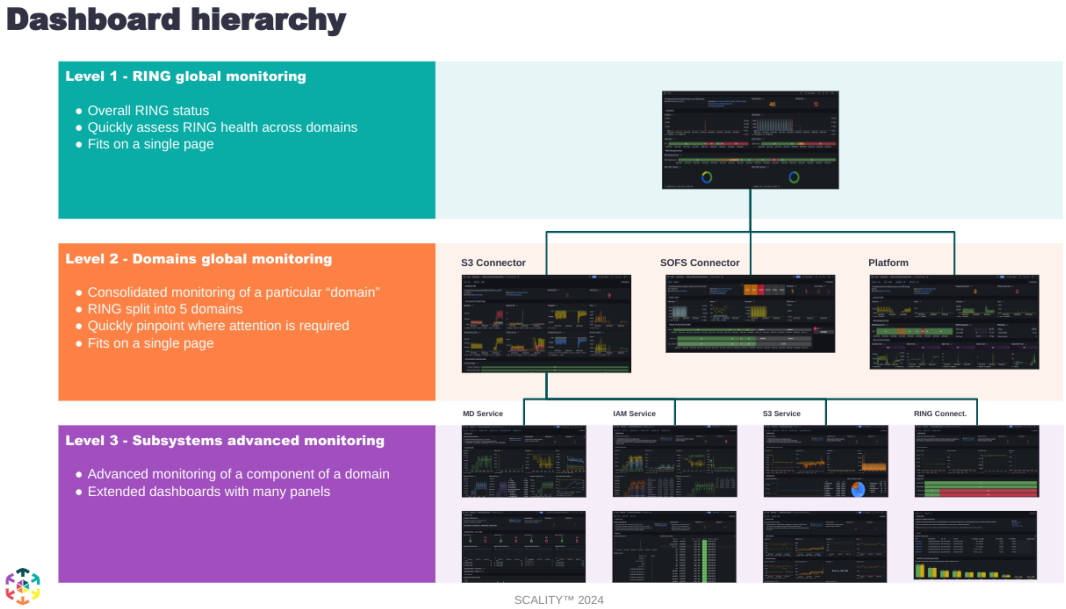
Pour la surveillance du RING à partir d'outils de centre de données courants tels que Nagios, le RING fournit une MIB compatible SNMP. Cela permet à ces outils de surveiller activement l'état du RING et de recevoir des alertes via des traps SNMP. Des statistiques sur l'état du système, la consommation de ressources, le connecteur et les statistiques de performances du nœud de stockage sont disponibles et peuvent être parcourues à partir de la MIB.

Au travers de Prometheus et de Grafana, nous obtenons l'écran d'accueil du superviseur ci-dessous après s'être authentifié par login et MFA par exemple est le suivant :

Pour disposer du dashboard de supervision complet, il suffit de sélectionner l'onglet "Monitoring" depuis le superviseur.



Les métriques sont larges et variées. Le passage en Prometheus+grafana apporte une modélisation différente du monitoring Scality RING. Il y a 3 niveaux de supervision : Global (vue générale), par domaine (santé du domaine, vue consolidée spécifique au domaine) et par sous-systèmes (vue avancée + troubleshooting).



Pour la partie "connecteur S3", le dashboard Grafana montre les métriques suivantes :

- Top Buckets

- Responses
- Latencies
- Operations
- Bandwidth

Il est également possible d'affiner les valeurs/graphiques au moyen des filtres mis en avant sur chaque dashboard : Host, Service, Layer, Operation, …

L'historisation des données (downsampling) est configurée par défaut et donc modifiable via le fichier de configuration de Prometheus. Exemple en standard pour Prometheus :

```
prometheus:
   downsampling:
      raw:
         retention: 2d
      level1:
         resolution: 5m
         retention: 10d
      level2:
         resolution: 1h
         retention: 6m
```

RESTful API

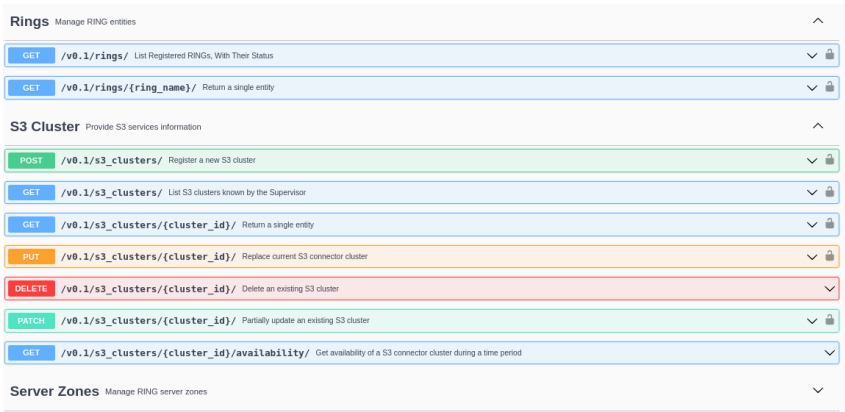
L'API RESTful de la solution utilise OpenAPI. L'API est «compatible avec Swagger» pour fournir une documentation de développement interactive, une génération de kits de développement logiciel (SDK) client et une possibilité de découverte.

L'API RESTful peut être utilisée pour :

- Identifier tous les serveurs, connecteurs, etc. connus du superviseur
- Utilisez cette liste pour implémenter un premier niveau de surveillance des composants RING (en utilisant l'état des composants).

Il offre :

- Prise en charge du contrôle d'accès basé sur les rôles (initialement 2 rôles, lecture seule et administrateur)
- Fonctions de surveillance et d'administration provenant d'applications tierces



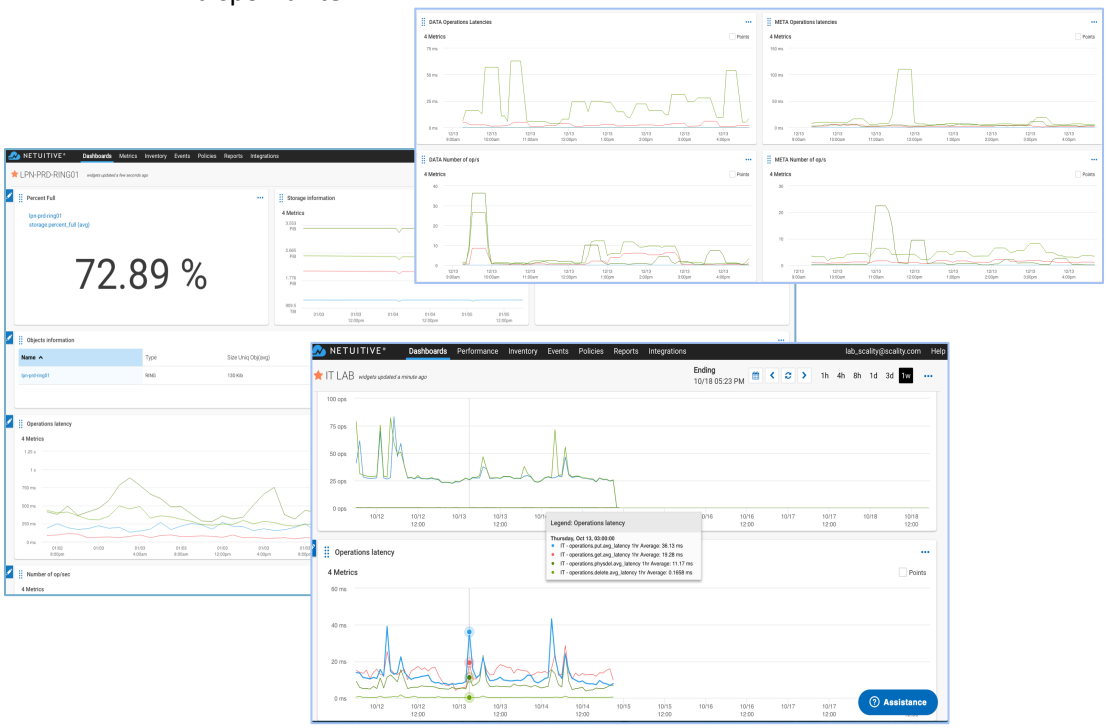
Scality Cloud Monitor

Dans le cadre d'une solution autorisée à communiquer automatiquement avec le Centre de Support

Scality (hors scope : “Dark site”, besoin d’upload depuis le superviseur Scality vers le cloud, porte 443 en outbound uniquement, métriques envoyées uniquement), Scality Cloud Monitor est une solution proactive de surveillance, d’alarme et de planification de la capacité. Scality Cloud Monitor fournit aux clients et au support Scality une vue sur l'état, les opérations, les performances et la santé du RING. Avec la version DCS (Dedicated Care Service) de RING, un service de surveillance complet est disponible dans le cadre du service Always-On pour les clients.
Les administrateurs peuvent voir des centaines de mesures, notamment le statut RING, la surveillance de la plate-forme / du disque, les nœuds, les démons de stockage, les connecteurs, les compartiments et les capacités de planification de la capacité. Scality Cloud Monitor DCS fournit également un système d’apprentissage et d’alerte proactif pour les métriques mentionnées, ce qui facilite la détection des problèmes et l’analyse des causes profondes.

Scality Cloud Monitor est basé sur 3 angles de surveillance:

- Analyse contextuelle et prédictive en temps réel - Grâce à un processus d'apprentissage basé sur l'analyse historique d'une énorme quantité de données, Scality Cloud Monitor prédit les valeurs attendues et le fonctionnement habituel du RING et de ses composants. Plusieurs types de performances des indicateurs clés (KPI) basés sur les comportements contextuels peuvent être configurés pour améliorer l'analyse des causes premières des problèmes et des opérations anormales. Les valeurs attendues des métriques peuvent également être prédites en fonction des modifications apportées à d'autres métriques.
- Vérifications de l’état des composants - Les sondes effectuent des vérifications individuelles des composants émulant la demande d’un client sur un serveur. Le code de retour du serveur (ou son absence) est analysé et s’il existe un écart par rapport au résultat attendu, une alarme est déclenchée.
- Moniteurs de service de bout en bout: ils sont conçus pour refléter «l'expérience de l'utilisateur final» en reproduisant un comportement typique de l'application. Les temps de sortie et de réponse sont mesurés, stockés et analysés. En règle générale, ces sondes peuvent être utilisées pour la surveillance des accords de niveau de disponibilité



Audit log

Toutes les actions effectuées dans l'interface du superviseur ou à l'aide d'une API REST sont consignées par défaut sur la machine hôte du superviseur. Le journal d'audit fournit le nom de l'utilisateur administrateur qui effectue une action (par exemple, ajout ou suppression d'un composant RING, enregistrement ou désinscription d'un serveur RING). Les journaux sont disponibles à des fins de support et de vérification. Pour les journaux d'accès S3, toutes les actions S3 et IAM sont connectées au système et sont disponibles pour être redirigées ou accédées via les serveurs ou via le composant ELK du Scality RING. Tous les événements peuvent être retrouvés jusqu'aux utilisateurs d'origine. Scality est compatible avec les journaux d'audit Linux (auditd) pour enregistrer les appels système par exemple, ouvrir un fichier, tuer un processus ou créer une connexion réseau et d'une manière plus générale tous les évènements relatifs à la plateforme. Les actions associées au RING sont tracées dans un log d'audit exploitable au format leef.

Les journaux d'événements sont capturés à l'aide de syslog et peuvent être envoyés dans un système central en UDP. Il n'y a pas de plug-in natif Splunk. Le logiciel "splunk universal forwarder" pourrait être installé sur les éléments du cluster sous réserve de compatibilité.

Les logs sont situés dans /var/log/scality/ sur chacun des serveurs scality.

Ci-dessous un exemple des logs du RING :
`salt/*.log` --> One log file per salt minions (e.g., master.log, supervisor.log)
`setup/installer.log` --> Log file that offers detail of all installation steps
`setup/*.log` --> One log file per installation step (e.g., bootstrap.log, install.log)

Le niveau de criticité des logs est le suivant :
criticalLogs requiring immediate attention
errorLogs requiring immediate attention, but which typically do not impact running environment
warningLogs that are important, but which may be considered later (typically, next business day)
infoLogs that reflect regular operations

/var/log/audit.log --> Fichier d'audit de l'interface graphique des actions faites sur le superviseur
/var/log/scality/supapi/audit.log --> Fichier d'audit de l'API des actions faites sur le superviseur
/var/log/scality/node --> Fichiers de logs des noeuds de stockage
/var/log/messages --> Contient les logs des informations disks, des informations de reconstruction de la donnée,

Tous les logs de la partie S3 se situe ici:
```
/var/log/s3/scality-s3/logs/s3-0.log
/var/log/s3/scality-bucketd/logs/bucketd-0.log
/var/log/s3/scality-vault/logs/vault-0.log
 /var/log/s3/scality-sproxyd/logs/nginx-0.log
```

Mise à jour du RING

Processus

Grâce à l'architecture peer-to-peer de Scality, la mise à niveau Scality (y compris les mises à niveau OS / firmware, et n'importe quelle maintenance sur les serveurs) peut être effectuer à chaud sans interruption de service, un serveur à la fois.
Pendant l'upgrade de tous les serveurs, le RING continuera à fonctionner avec un environnement mixte, constitué de serveurs avec des versions différentes d'OS différentes.

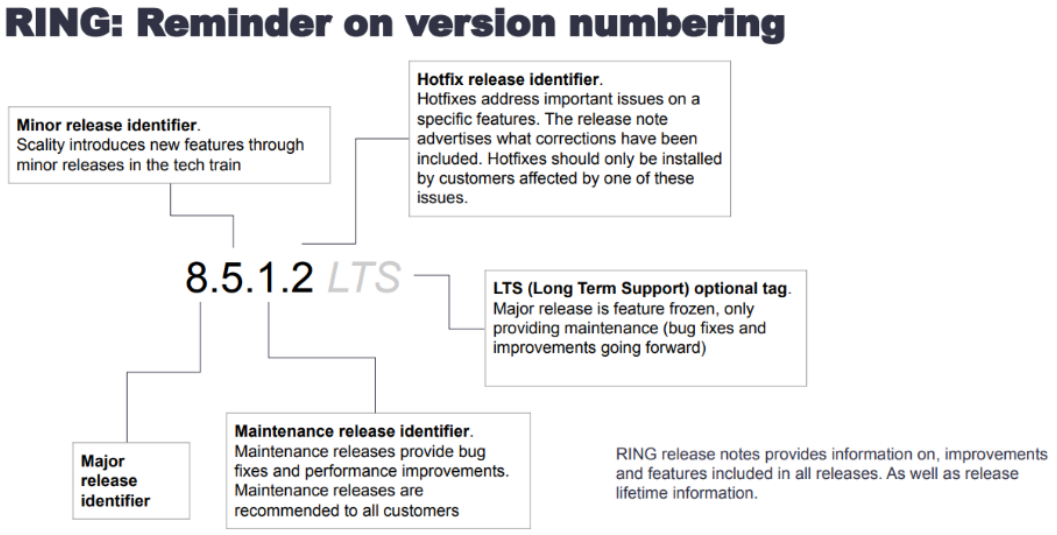
Le processus de mise à niveau du logiciel Scality consiste à mettre à niveau les packages Scality à partir du référentiel mis à jour. Pour maintenir la production, il est nécessaire de mettre à niveau les serveurs, un à la fois, en attendant que le serveur soit à nouveau complètement actif avant de mettre à niveau le serveur suivant. Cela se fait au niveau du superviseur, du serveur de stockage (nœuds) et du connecteur lors de la mise à niveau des packages.

- Superviseur - Le superviseur est un composant passif qui peut être mis à niveau sans impact sur le RING.
- Storenodes - Lorsqu'un serveur de stockage est indisponible (lors de sa mise à niveau), toute modification apportée aux objets de l'espace de clés de ce serveur sera temporairement ignorée. Les mises à jour de ces objets seront propagées ultérieurement et automatiquement par la tâche de reconstruction. Les nouveaux objets entrant seront écrits sur d'autres serveurs. Une fois la mise à niveau terminée et le serveur rétabli, le Ring va s'adapter aux modifications (la tâche en arrière-plan est effectuée pour obtenir l'équilibre).
- Connecteurs - les nouveaux connecteurs seront enregistrés après la livraison du paquet.

Le processus de mise à jour est décrit dans la documentation Scality RING et peut différer suivant les versions et les fonctionnalités insérées au fur et à mesure de la vie du produit.

Fréquence & roadmap

Scality Ring a mis en place la nomenclature ci-dessous pour ses releases software :
X = Version majeure, Y = Version mineure, Z = Numéro de version du patch.



Scalité est passée d'une méthodologie de publication majeure à un modèle de fonctionnalités en continu. Cela signifie que les nouvelles fonctionnalités seront désormais livrées dans des versions

mineures plus fréquentes tous les 3 mois environ. La disponibilité générale (GA) signifie la livraison au fur et à mesure que les fonctionnalités deviennent complètes et testées. Chaque année, cette version est complétée par une version LTS (Long Term Supported) et la prochaine révision du logiciel est une version majeure de X.0 destinée à lancer la nouvelle version de X.1, X.2, etc. minimum 4 ans, 2 ans de maintenance active (correctifs proactifs) + 2 ans de maintenance prolongée (patchs au besoin).
Enfin, toutes les versions prennent en charge les mises à niveau continues et en ligne.
Actuellement, la LTS RING est 8.5.8 tandis que la Tech Train est 9.4 avec une 9.5 prévue dans les prochains mois.
La roadmap quant à elle implique des nouvelles fonctionnalités sur la Tech Train 9.5 et suivantes ainsi que la release majeur RING 10.

Mode de licencing

Le licensing du logiciel RING s'effectue au TB usable et protégé présent dans le cluster. Il est possible ainsi d'évoluer de façon granulaire en fonction des besoins. La licence n'a pas d'impact sur la Production et n'engendre pas d'interruption de service lors de sa mise à jour.