



UNIVERSITÉ SAINT JEAN
SAINT JEAN INGÉNIEUR

SÉCURITÉ DU CLOUD

THÈME : Stockage objet évolutif avec Swift

Réalisé par :

- FOUDA DZOU AMBE Valdez
- KAMDOUM TEUFENG Josette

Sous la coordination de : Mr. NGUIMBUS Emmanuel

Année Académique
2025-2026

TABLE DE MATIERES

INTRODUCTION.....	1
I. CONTEXTE.....	3
II. DEFINITION DU PROBLEME.....	3
III. OBJECTIFS VISÉS.....	3
IV. DESCRIPTION DES FONCTIONNALITÉS.....	4
V. RÉSULTAT ATTENDU.....	5
1. C'est quoi une Api ?.....	5
2. C'est quoi Swift ?.....	6
3. Présentation de l'application Web.....	7
VI. RESSOURCES UTILISÉES.....	8
1. Ressources Matérielles.....	8
2. Ressources Logicielles.....	8
3. Ressources Humaines.....	9
CONCLUSION.....	10

INTRODUCTION

Ce cahier des charges présente les spécifications du projet de développement d'une application web innovante conçue pour automatiser la sauvegarde des données vers une solution de stockage objet évolutive et hautement disponible basée sur OpenStack Swift. Cette plateforme, reconnue pour sa capacité à gérer de manière sécurisée et scalable de très grands volumes de données non structurées, constitue une base robuste pour répondre aux enjeux modernes de gestion des données. L'application intégrera des fonctionnalités telles que le versionning des objets, permettant de conserver et restaurer différentes versions des données, et la réplication inter-sites, assurant une haute disponibilité et une résilience face aux pannes. De plus, le projet sera réalisé avec Angular pour la partie front-end, offrant une interface utilisateur intuitive et réactive, facilitant ainsi la gestion des sauvegardes et la supervision des données pour les utilisateurs finaux.

I. CONTEXTE

Avec l'explosion constante des volumes de données générées par les entreprises et les organisations, il devient impératif de mettre en place des solutions de stockage capables de suivre cette croissance de manière efficace. Les systèmes traditionnels basés sur le stockage par fichiers ou par blocs montrent rapidement des limites en ce qui concerne la scalabilité, la disponibilité, ainsi que le coût d'exploitation. Ces solutions peinent à s'adapter aux exigences actuelles de flexibilité et de robustesse imposées par les environnements numériques modernes. C'est dans ce cadre que le stockage objet, et plus particulièrement la solution OpenStack Swift, s'impose comme une alternative innovante. OpenStack Swift est une plateforme de stockage distribuée, conçue pour gérer de très grandes quantités de données non structurées tout en offrant une tolérance aux pannes élevées. Le présent projet se propose d'exploiter cette technologie pour concevoir une application de sauvegarde automatique des données, intégrant un mécanisme avancé de versionning et une réplication inter-sites afin d'assurer une haute disponibilité des objets stockés.

II. DEFINITION DU PROBLEME

Le principal défi auquel sont confrontées les entreprises aujourd'hui consiste à gérer des volumes de données numériques en constante augmentation, sans compromettre la sécurité, la disponibilité, ni la performance. Les solutions traditionnelles ne répondent plus à ces besoins, en particulier lorsqu'il s'agit de données non structurées telles que les fichiers multimédias, documents, ou autres contenus divers. Un système de sauvegarde fiable, capable de garantir la conservation de toutes les versions des données, et capable par ailleurs d'assurer une continuité de service même en cas de panne majeure, est donc indispensable. Ce projet vise à combler cette lacune en développant une solution s'appuyant sur OpenStack Swift, offrant ainsi un stockage évolutif, répliqué entre plusieurs sites pour garantir la résilience et la disponibilité des données sauvegardées.

III. OBJECTIFS VISÉS

L'objectif principal est de développer une application capable d'automatiser la sauvegarde des données vers OpenStack Swift avec une prise en charge complète du versionning. Cette application devra permettre de restaurer n'importe quelle version antérieure d'un objet sauvegardé, offrant ainsi une protection maximale contre les pertes ou altérations accidentelles. Parallèlement, le projet doit mettre en place un mécanisme de réplication inter-sites efficace, permettant de synchroniser les données entre plusieurs clusters Swift géographiquement distants. Ceci garantira que les données restent accessibles en toute circonstance, y compris lors d'un incident affectant un centre de données. Cette architecture sécurisée et distribuée vise à assurer une scalabilité optimale, une haute disponibilité, et une robustesse face aux défaillances matérielles ou logicielles.

Les objectifs principaux du projet sont :

1. Mettre en place une infrastructure Swift fonctionnelle et distribuée pour le stockage d'objets.
2. Développer une application de sauvegarde automatique permettant d'envoyer les données locales vers Swift.
3. Intégrer un mécanisme de versioning, permettant de conserver l'historique des modifications.
4. Configurer une réplication inter-sites, afin d'assurer la haute disponibilité et la résilience des données.
5. Tester et valider le bon fonctionnement du système (sauvegarde, restauration, basculement).

IV. DESCRIPTION DES FONCTIONNALITÉS

Pour répondre aux besoins identifiés dans ce projet, l'application web développée aura pour objectif principal de fournir une plateforme complète et conviviale de gestion du stockage objet avec OpenStack Swift. Elle devra non seulement automatiser la sauvegarde des données locales vers la solution Swift tout en gérant intelligemment les versions des objets stockés, mais aussi permettre la supervision en temps réel de la réplication entre sites distants. Cette approche garantit une haute disponibilité et une robustesse des données sauvegardées.

L'interface utilisateur de l'application sera conçue pour offrir une expérience simple et intuitive, facilitant la configuration des paramètres de sauvegarde, la gestion sécurisée des accès, et l'accès aux différentes fonctionnalités de contrôle. Les fonctionnalités intégrées auront pour but d'accompagner efficacement les utilisateurs dans leurs tâches de sauvegarde tout en assurant la fiabilité et la continuité des services.

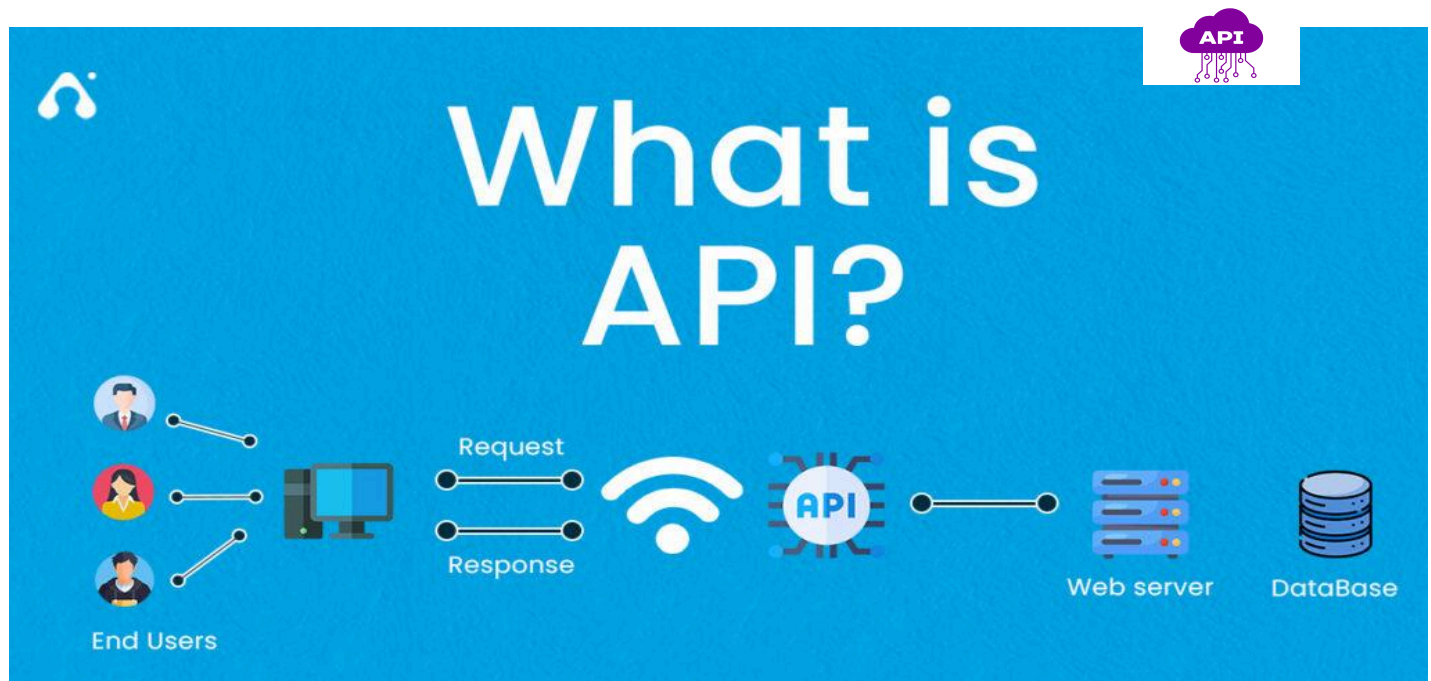
FONCTIONNALITÉS	DESCRIPTIONS
Interface utilisateur simple et intuitive pour configurer les paramètres de sauvegarde.	Une interface claire permettant aux utilisateurs de définir facilement les paramètres de sauvegarde, tels que la fréquence, les dossiers ciblés, et les options de versionning.
Gestion des comptes et accès sécurisés aux clusters Swift.	Mise en place d'un système d'authentification et d'autorisation garantissant que seuls les utilisateurs habilités peuvent accéder et manipuler les données stockées.
Automatisation de la sauvegarde périodique des fichiers/données locales vers Swift.	Fonctionnalité permettant de programmer et d'exécuter automatiquement des sauvegardes régulières sans intervention manuelle.
Affichage et gestion des versions multiples de chaque objet sauvegardé.	Possibilité de visualiser les différentes versions sauvegardées d'un même fichier et de gérer ces versions pour restauration ou suppression.
Visualisation du statut de réplication des données entre sites.	Suivi en temps réel de la réplication des données sur plusieurs sites pour assurer la continuité et la disponibilité.
Fonctionnalité de restauration des données à partir de versions sauvegardées.	Interface permettant de restaurer facilement une version antérieure d'un objet en cas de besoin.

Notifications et alertes en cas d'échec ou d'incident.	Système d'alerte pour informer rapidement les utilisateurs en cas de problème lors des sauvegardes ou de la réplication.
Tableau de bord statistique sur les volumes stockés et la santé des clusters.	Présentation visuelle des données statistiques clés sur l'utilisation du stockage et les performances des clusters Swift.

V. RÉSULTAT ATTENDU

1. C'est quoi une Api ?

Une API (Application Programming Interface), ou interface de programmation d'application, est un ensemble de règles, protocoles et outils permettant à des logiciels ou applications de communiquer entre eux. Elle définit comment un programme peut interagir avec un autre en envoyant des demandes (requêtes) et en recevant des réponses, sans avoir besoin de connaître les détails internes de ce dernier. En résumé, une API sert de pont qui facilite l'échange de données et l'utilisation de fonctionnalités d'un logiciel par un autre, ce qui permet de construire des applications plus modulaires et collaboratives.

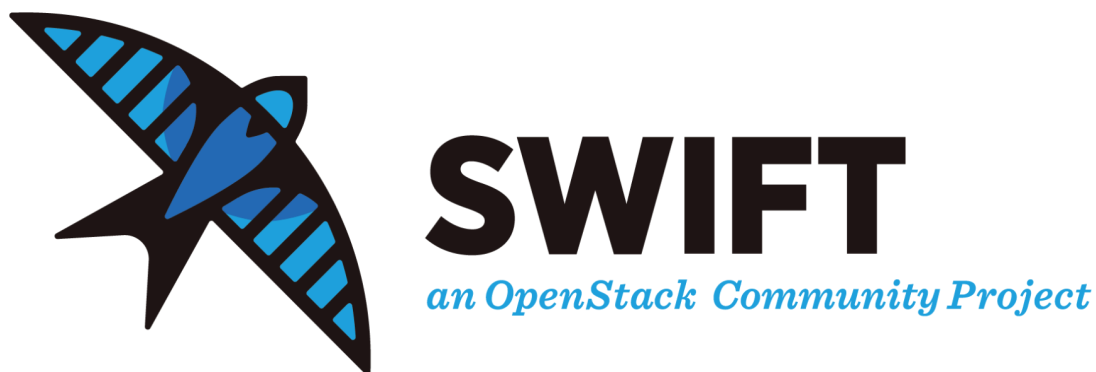


2. C'est quoi Swift ?

OpenStack Swift est un système de stockage d'objets distribué et tolérant aux pannes, conçu pour stocker de grandes quantités de données non structurées de manière scalable,

allant d'un seul serveur à des milliers. Il est optimisé pour la multi-location et la haute concurrence, ce qui le rend idéal pour les sauvegardes, le contenu web et mobile, ou tout type de données non structurées qui peuvent croître sans limite.

Swift fonctionne en répliquant les données sur plusieurs nœuds dans un cluster, assurant ainsi la résilience et la continuité d'accès même en cas de défaillance de certains serveurs. L'accès aux données se fait via une interface REST simple. Il est souvent utilisé comme base pour des solutions cloud privées ou publiques, offrant une gestion évolutive, fiable et sécurisée du stockage d'objets.



3. Présentation de l'application Web

L'application web développée en **Angular** sert d'interface utilisateur frontale pour interagir avec la solution de stockage objet OpenStack Swift, qui expose des API RESTful compatibles. Ces API permettent à l'application de gérer les objets stockés, les containers, les comptes utilisateurs, ainsi que les opérations de sauvegarde, de versionning et de réplication.

Grâce à ces API, l'application n'a pas besoin de gérer directement le stockage physique des données mais agit comme un client consommateur des services Swift pour envoyer, récupérer, et manipuler les objets stockés. Cette architecture découple clairement le front-end Angular des complexités du backend Swift tout en garantissant une intégration fluide et sécurisée.

SCÉNARIOS	DESCRIPTIONS
Sauvegarde automatisée des fichiers	L'utilisateur sélectionne un ou plusieurs fichiers dans l'interface. L'application Angular utilise l'API Swift pour créer un container si besoin, puis envoie les fichiers dans ce container via des requêtes POST ou PUT. Le versionning est activé en

	ajoutant des métadonnées versionnées à chaque objet.
Consultation des versions d'un objet	L'utilisateur visualise l'historique des versions pour un fichier donné. L'application interroge l'API Swift pour lister les versions disponibles stockées dans le container et affiche les options de restauration.
Restauration d'une version antérieure	En sélectionnant une version, l'utilisateur déclenche une requête GET via l'API pour télécharger la version désirée. Le fichier peut alors être restauré soit localement, soit remplacé comme version la plus récente.
Suivi de la réplication inter-sites	L'application interroge les API Swift pour obtenir le statut de réplication des données entre différents clusters géographiquement distants et affiche ces informations dans le tableau de bord.
Gestion sécurisée des comptes	L'application gère l'authentification des utilisateurs via des API d'identité (ex : Keystone dans OpenStack) pour contrôler et limiter les accès aux données selon les permissions attribuées.

VI. RESSOURCES UTILISÉES

1. Ressources Matérielles

RESSOURCES	DESCRIPTIONS	COÛTS
05 Machines Virtuelles	<p>Machines virtuelles pour héberger :</p> <ul style="list-style-type: none"> Le cluster OpenStack Swift (nœuds de stockage, proxy server, etc.) Les services d'identité (Keystone) 	50.000 FCFA/mois pour 6 mois soit 300.000 FCFA

	<ul style="list-style-type: none"> Le serveur d'application backend (si besoin) 	
Connexion Internet	haut débit	25.000 FCFA/mois soit 250.000 FCFA
10 Ordinateurs Portables	Capacités : 12G RAM 256G SSD 2.1GHz	2.500.000 FCFA (250.000 FCFA l'Unité)
	TOTAL :	3.050.000 FCFA

2. Ressources Logicielles

RESSOURCES	DESCRIPTIONS	COÛTS
OpenStack Swift	(service de stockage objet) installé sur le cluster	0
OpenStack Keystone	service d'authentification et d'autorisation	0
Framework Angular	pour le développement front-end	0
Environnement backend	Node.js	0
Outils de virtualisation ou conteneurs	Docker, pour tests et déploiement	0
Serveur web et infrastructure pour déployer l'application Angular	Nginx	0
Outils de gestion de versions	Git, GitHub/GitLab	0
	TOTAL :	0 FCFA

3. Ressources Humaines

RESSOURCES	COÛTS
05 - Développeurs front-end maîtrisant Angular et les technologies web modernes (HTML, CSS, TypeScript)	100.000 FCFA/mois pour chaque développeur soit 3.000.000 FCFA

05 - Développeurs backend (si architecture multi-tier) connaissant les API REST, sécurité et intégration avec OpenStack	100.000 FCFA/mois pour chaque développeur soit 3.000.000 FCFA
01 - Expert en sécurité informatique pour configurer les accès et la gestion des permissions	500.000 FCFA
01 - Chef de projet pour coordination, planification et suivi des développements	500.000 FCFA
TOTAL :	7.000.000 FCFA

CONCLUSION

Ce projet répond à un besoin essentiel des entreprises pour gérer efficacement et en toute sécurité de grandes quantités de données non structurées. Grâce à OpenStack Swift, il offre un stockage évolutif, fiable et redondant, avec des mécanismes avancés de sauvegarde automatique, de versionning et de réplication inter-sites. Cette solution garantit la continuité d'activité même en cas de panne, sécurisant ainsi les actifs numériques stratégiques des organisations. De plus, l'application Angular apporte une interface accessible et intuitive, facilitant son adoption par les équipes opérationnelles et améliorant la gestion quotidienne des sauvegardes et restaurations.