**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**MÔN HỌC: QUẢN TRỊ MẠNG VÀ HỆ THỐNG**

**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**TÌM HIỂU VÀ TRIỂN KHAI HỆ THỐNG S3 VỚI TÍNH SẴN SÀNG CAO (MinIO)**

**Giảng viên hướng dẫn: Nguyễn Duy**

**NT132.O11.ANTN**

**Sinh viên thực hiện:**

**21522056 – Hà Thị Thu Hiền**

**21522641 – Phạm Ngọc Thơ**

**21521248 – Nguyễn Ngọc Nhung**

**TP.HCM, ngày 30 tháng 11 năm 2023**

Contents

[**A.** **BÁO CÁO CHI TIẾT** 2](#_Toc152969208)

[I.Mở đầu 2](#_Toc152969209)

[1. Giới thiệu 2](#_Toc152969210)

[2. Cơ sở lý thuyết 3](#_Toc152969211)

[3. Phân tích so sánh 3](#_Toc152969212)

[4. Triển khai MinIO 5](#_Toc152969213)

[5. Các feature trong MinIO 7](#_Toc152969214)

[6. Tìm hiểu thêm về MinIO Feature Overview: Object Lifecycle Management 31](#_Toc152969215)

# **BÁO CÁO CHI TIẾT**

## I.Mở đầu

### 1. Giới thiệu

* Đề tài của chúng ta là “Tìm hiểu và triển khai hệ thống S3 với tính sẵn sàng cao”. Trong thế giới số hóa ngày nay, việc lưu trữ và truy cập dữ liệu một cách nhanh chóng, an toàn và tin cậy là vô cùng quan trọng. Amazon S3, một dịch vụ lưu trữ đối tượng phổ biến, đã đặt ra tiêu chuẩn cho việc này. Tuy nhiên, việc triển khai và duy trì một hệ thống S3 có thể đòi hỏi nhiều nguồn lực và kỹ năng chuyên môn.
* Mục tiêu của đề tài này là tìm hiểu về MinIO, một giải pháp lưu trữ đối tượng mã nguồn mở tương thích với Amazon S3, và cách triển khai nó để đạt được tính sẵn sàng cao. MinIO không chỉ cung cấp một giải pháp lưu trữ đối tượng hiệu quả và linh hoạt, mà còn có thể được triển khai trên nhiều môi trường khác nhau, từ máy chủ đơn lẻ đến các cụm lớn.
* Phạm vi của đề tài này bao gồm việc tìm hiểu về MinIO, cài đặt và cấu hình nó, và cuối cùng là kiểm tra hiệu suất và độ tin cậy của nó. Chúng ta sẽ bắt đầu bằng việc tìm hiểu về MinIO và cách nó hoạt động, sau đó chúng ta sẽ đi sâu vào việc cài đặt và cấu hình MinIO. Cuối cùng, chúng ta sẽ kiểm tra hiệu suất và độ tin cậy của hệ thống đã triển khai.
* Phương pháp nghiên cứu của chúng ta bao gồm việc tìm hiểu lý thuyết, thực hành và phân tích kết quả. Chúng ta sẽ tìm hiểu về MinIO thông qua các tài liệu hướng dẫn và tài liệu tham khảo từ cộng đồng. Chúng ta sẽ thực hành bằng cách cài đặt và cấu hình MinIO trên một môi trường thử nghiệm. Cuối cùng, chúng ta sẽ phân tích hiệu suất và độ tin cậy của hệ thống đã triển khai bằng cách thực hiện các bài kiểm tra và đánh giá kết quả.

### 2. Cơ sở lý thuyết

#### Khái niệm S3

* Amazon Simple Storage Service (S3) là một dịch vụ lưu trữ đối tượng được cung cấp bởi Amazon Web Services (AWS). S3 được thiết kế để lưu trữ và truy xuất bất kỳ loại dữ liệu nào từ bất kỳ nơi nào trên web. Nó cung cấp một giao diện lập trình ứng dụng (API) đơn giản cho người dùng để lưu trữ và truy xuất dữ liệu.

#### Nguyên lý hoạt động của S3

* Trong S3, dữ liệu được lưu trữ dưới dạng đối tượng trong các “bucket”. Mỗi đối tượng bao gồm dữ liệu và các metadata mô tả dữ liệu. Metadata có thể bao gồm các thông tin như tên đối tượng, kích thước, ngày tạo, và các thông tin khác.
* Bucket hoạt động như một thư mục, cho phép người dùng tổ chức các đối tượng theo cách của họ. Mỗi bucket phải có tên duy nhất trong không gian tên S3.

#### Yếu tố ảnh hưởng đến tính sẵn sàng cao của S3

* Tính sẵn sàng cao của hệ thống S3 phụ thuộc vào nhiều yếu tố, bao gồm cấu hình hạ tầng, quản lý và giám sát, và việc sử dụng các công nghệ như erasure coding và multi-site replication.
* Cấu hình hạ tầng: Việc chọn hạ tầng phù hợp, bao gồm số lượng và loại máy chủ, mạng, và lưu trữ, có thể ảnh hưởng đến tính sẵn sàng cao của hệ thống S3.
* Quản lý và giám sát: Việc quản lý và giám sát hệ thống S3 một cách chặt chẽ, bao gồm việc theo dõi hiệu suất, phát hiện và khắc phục sự cố, cũng có thể ảnh hưởng đến tính sẵn sàng cao.
* Erasure coding và multi-site replication: Erasure coding là một phương pháp mã hóa dữ liệu sao cho dữ liệu có thể được khôi phục ngay cả khi một phần của nó bị mất. Multi-site replication là việc sao chép dữ liệu giữa nhiều vị trí để tăng tính sẵn sàng và độ tin cậy.

### 3. Phân tích so sánh

|  |  |  |
| --- | --- | --- |
| 7 Best Self-Hosted S3-Compatible Object Storage Software | Ưu điểm | Nhược điểm |
| MinIO | - Hiệu suất cao, có khả năng đọc/viết nhanh.  - Hỗ trợ nhiều trường hợp sử dụng khác nhau.  - Hỗ trợ các công nghệ và kiến trúc cloud.  - Open-source, giúp giảm lỗi và tăng tính ổn định.  - Dễ cài đặt và cấu hình. | Yêu cầu tự quản lý và bảo trì. |
| Ceph | - S3 tương thích và hỗ trợ các giao diện OpenStack Swift.  - Hỗ trợ nhiều ngôn ngữ lập trình.  - Các tính năng nâng cao như snapshots, key-value mappings, giao dịch atomic. | Cần kiến thức kỹ thuật trong việc cài đặt và quản lý. |
| Zenko | - Hỗ trợ giao diện S3, cho phép lưu trữ dữ liệu ở nhiều nơi.  - Hỗ trợ nhiều ngôn ngữ lập trình và SDK.  - Giao diện đơn giản cho việc tích hợp ứng dụng. | Dựa vào các dịch vụ công cộng và cần kết nối mạng ổn định. |
| Riak S2 | - Tương thích S3 và OpenStack Swift.  - Hỗ trợ nhiều tính năng và thư viện cho các ứng dụng lớn.  - Tích hợp công cụ dễ sử dụng cho nhà phát triển. | Cần kiến thức kỹ thuật để cấu hình và quản lý. |
| Triton | - Giao diện đơn giản và dễ sử dụng.  - Bảo mật mạnh mẽ và khả năng điều khiển tốt.  - Có khả năng mở rộng và tái tạo dữ liệu tốt. | Yêu cầu sử dụng giao diện dòng lệnh Unix. |
| LeoFS | - Lưu trữ lớn và hiệu suất tốt.  - Có khả năng mở rộng dễ dàng.  - Độ tin cậy cao và thời gian hoạt động liên tục. | Yêu cầu quản lý và bảo trì tự quản lý. |
| HyperStore (bởi Cloudian) | - Hỗ trợ nhiều loại nền tảng và cấu hình.  - S3 tương thích và hỗ trợ NFS & SMB.  - Có khả năng mở rộng và tích hợp đám mây. | Phải trả phí sau thời gian thử nghiệm. |

Tham khảo: [7 Best Self-Hosted S3-Compatible Object Storage Software (geekflare.com)](https://geekflare.com/self-hosted-s3/)

### 4. Triển khai MinIO

Mô hình: MinIO là một dịch vụ lưu trữ đối tượng phân tán được viết bằng ngôn ngữ Go và mã nguồn mở dưới bản quyền Apache phiên bản 2.01. MinIO được thiết kế để cung cấp các tỷ lệ truyền thông cao nhất cho các nhiệm vụ làm việc với dữ liệu lớn, như Big Data, IoT và AI2. MinIO cũng hỗ trợ các tính năng như namespace, encryption, compression, deduplication và erasure coding.

A diagram of a computer network

Description automatically generated

* Component: MinIO gồm hai thành phần chính: server và client. Server là nơi lưu trữ và xử lý các đối tượng (object) trong hệ thống. Client là nơi giao tiếp với server để đọc hoặc ghi các đối tượng. Các thành phần khác bao gồm các thành phần phụ thuộc như storage controller, network controller, erasure set manager và bucket manager.
* Performance: MinIO có hiệu năng cao do sử dụng các công nghệ tiên tiến như NVMe SSDs cho khả năng xử lý cao (tối thiểu 8 ổ cứng cho máy chủ), SATA/SAS HDDs cho dung lượng và 25GbE cho dung lượng và 100GbE NICs cho hiệu năng cao. MinIO cũng sử dụng erasure coding để tái tổ chức các đối tượng thành các shard data và parity để duy trì tính toàn vẹn của dữ liệu khi có sự cố xảy ra. MinIO có thể chạy được hàng triệu truy vấn Spark\*, Presto\* và Hive\* hoặc thực hiện nhanh các thuật toán AI. MinIO có hiệu suất cao nhất trong ngành với kết quả GETs/PUTs công bố đã vượt qua 325 GiB/sec và 165 GiB/sec khi triển khai trên cluster 32 node của NVMe drive và mạng 100Gbe. MinIO cũng có hiệu suất cao khi sử dụng NVMe drive hoặc SSD drive gắn vào PCI-E controller board trong host machine.
* High availability: MinIO có khả năng duy trì sự khả dụng cao do sử dụng erasure coding để sao lưu toàn bộ hoặc một phần của dữ liệu vào các node khác trong pool deployment. Nếu có node yếu hoặc không hoạt động được, node khác sẽ chịu trách nhiệm xử lý các yêu cầu từ client. Ngoài ra, MinIO cũng hỗ trợ backup and restore để khôi phục lại dữ liệu từ backup storage.
* Security: MinIO có tính bảo mật cao do sử dụng encryption để mã hóa toàn bộ hoặc một phần của dữ liệu khi gửi qua network. Encryption được thực hiện bởi client hoặc server theo yêu cầu của client. Ngoài ra, MinIO cũng hỗ trợ authentication và authorization để kiểm soát quyền truy cập vào server và bucket. MinIO còn có tường lửa bảo vệ giữa server và client.
* Operation: MinIO có tính linh hoạt cao do sử dụng namespace để phân biệt giữa các bucket con trong pool deployment. Namespace giúp người dùng có thể quản lý được không gian lưu trữ riêng biệt cho từng ứng dụng hoặc nhóm người dùng. Ngoài ra, MinIO cũng hỗ trợ replication để sao chép lại toàn bộ hoặc một phần của bucket sang bucket khác trong pool deployment hoặc sang remote storage.

#### Cài đặt MinIOn cho cụm máy

* Đầu tiên, chúng ta cần cài đặt MinIO trên một cụm máy chủ. Cụ thể, chúng ta cần tải xuống và cài đặt phần mềm MinIO từ trang web chính thức của MinIO. Sau khi cài đặt, chúng ta cần khởi động MinIO và kiểm tra xem nó có hoạt động đúng không.

#### Cấu hình MinIO cho cụm máy

* Tiếp theo, chúng ta cần cấu hình MinIO để hoạt động theo chế độ distributed mode. Điều này đòi hỏi chúng ta phải chọn số lượng node, loại node (CPU hoặc GPU), loại ổ (JBOD hoặc NVMe), loại erasure coding (0 erasure coding hoặc 1 erasure coding), và loại load balancer (NGINX hoặc HAProxy).
* Số lượng node: Số lượng node cần phải đủ lớn để đảm bảo tính sẵn sàng cao, nhưng không quá lớn để tránh lãng phí tài nguyên. Một số lượng node phổ biến là từ 4 đến 16.
* Loại node (CPU hoặc GPU): Loại node phụ thuộc vào yêu cầu hiệu suất của hệ thống. Nếu hiệu suất là ưu tiên hàng đầu, chúng ta nên chọn node GPU. Ngược lại, nếu chi phí là ưu tiên hàng đầu, chúng ta nên chọn node CPU.
* Loại ổ (JBOD hoặc NVMe): Loại ổ cũng phụ thuộc vào yêu cầu hiệu suất của hệ thống. Nếu hiệu suất là ưu tiên hàng đầu, chúng ta nên chọn ổ NVMe. Ngược lại, nếu chi phí là ưu tiên hàng đầu, chúng ta nên chọn ổ JBOD.
* Loại erasure coding (0 erasure coding hoặc 1 erasure coding): Erasure coding là một kỹ thuật mã hóa dữ liệu giúp khôi phục dữ liệu khi một phần của nó bị mất. Nếu tính sẵn sàng là ưu tiên hàng đầu, chúng ta nên chọn 1 erasure coding. Ngược lại, nếu hiệu suất là ưu tiên hàng đầu, chúng ta nên chọn 0 erasure coding.
* Loại load balancer (NGINX hoặc HAProxy): Load balancer giúp phân phối yêu cầu đến các node trong cụm. Nếu hiệu suất là ưu tiên hàng đầu, chúng ta nên chọn HAProxy. Ngược lại, nếu tính linh hoạt và dễ dàng cấu hình là ưu tiên hàng đầu, chúng ta nên chọn NGINX.
* Cuối cùng, chúng ta cần kết nối các node qua DNS hostname hoặc IP address. Điều này đòi hỏi chúng ta phải cấu hình DNS hoặc IP tĩnh cho mỗi node.

#### Kiểm tra hiệu suất và độ tin cậy

* Sau khi cài đặt và cấu hình MinIO, chúng ta cần kiểm tra hiệu suất và độ tin cậy của nó. Chúng ta có thể thực hiện điều này bằng cách tạo ra một số lượng lớn yêu cầu đọc và ghi đồng thời, và sau đó đo lường thời gian phản hồi và tỷ lệ lỗi. Chúng ta cũng nên kiểm tra tính sẵn sàng của hệ thống bằng cách tắt một số node và xem hệ thống có thể khôi phục không.

#### Cài đặt và cấu hình MinIOn cho 1 máy

[minio/minio: High Performance Object Storage for AI (github.com)](https://github.com/minio/minio/)

### 5. Các feature trong MinIO

* **Versioning (Phiên bản):**
  + Use Case: Người dùng muốn duy trì lịch sử phiên bản của các đối tượng trong bucket để có khả năng phục hồi dữ liệu nếu cần thiết.
  + Business Case: Giữ lại phiên bản trước của các đối tượng để có khả năng quản lý lỗi, phục hồi dữ liệu hoặc thực hiện kiểm tra và so sánh phiên bản.

Test Case:

* **Test Step 1: Tạo bucket với phiên bản:**
  + Sử dụng giao diện hoặc API để tạo một bucket mới với tính năng versioning được kích hoạt.
* **Test Step 2: Tạo và xoá đối tượng với phiên bản:**
  + Tạo một đối tượng trong bucket.
  + Thực hiện một số sửa đổi trên đối tượng và xác nhận rằng phiên bản mới được tạo.

A screenshot of a computer

Description automatically generated

A white background with a black and white background

Description automatically generated with medium confidenceA white rectangular object with black text

Description automatically generatedA white background with a black border

Description automatically generated with medium confidence

* + Xoá đối tượng và xác nhận rằng phiên bản cũ được giữ lại.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + Lỗi “Object, ‘input.py (Version ID=735f2082-f067-4d9b-8680-faaeb105e5c0)’ is WORM protected and cannot be overwritten” có nghĩa là đối tượng ‘input.py’ với ID phiên bản ‘735f2082-f067-4d9b-8680-faaeb105e5c0’ được bảo vệ bởi chính sách WORM (Write Once Read Many) và không thể bị ghi đè.
  + Chính sách WORM là một chính sách lưu trữ dữ liệu phổ biến, đặc biệt trong các ngành công nghiệp cần tuân thủ quy định như tài chính và y tế. Theo chính sách này, một khi dữ liệu đã được ghi, nó không thể bị sửa đổi hoặc xóa trong một khoảng thời gian nhất định hoặc cho đến khi một sự kiện nhất định xảy ra.
* **Test Step 3: Kiểm tra và khôi phục phiên bản:**
  + Kiểm tra danh sách các phiên bản của đối tượng.
  + Thử khôi phục phiên bản cũ và xác nhận rằng đối tượng được phục hồi đúng phiên bản cũ.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Như vậy thì curent version đã là phiên bản cũ mà ta đã restore, và nó tạo thành version v3 (current version).

- Cụ thể hơn ta sẽ kiểm tra trong code của input.py của từng version v1 v2 v3.

- input.py v1

A computer code with black text

Description automatically generated with medium confidence

* + input.py v2

A computer code with black text

Description automatically generated

* + Input.py v3 (current version) sau khi restore v1

A computer code with black text

Description automatically generated with medium confidence

* **Object Locking (Khóa đối tượng):**
  + Use Case: Người dùng muốn đảm bảo rằng các đối tượng không thể bị sửa đổi hoặc xóa trong một khoảng thời gian cụ thể.
  + Business Case: Bảo vệ dữ liệu quan trọng trước các thay đổi không được phép trong một khoảng thời gian xác định, đặc biệt là trong trường hợp yêu cầu luật pháp hoặc quy định.

Test Case:

* **Test Step 1: Khóa và thử sửa đổi đối tượng:**
  + Khóa một đối tượng trong bucket.
  + Thử thực hiện sửa đổi trên đối tượng và xác nhận rằng việc sửa đổi bị từ chối do đối tượng đã bị khóa.

A screenshot of a computer

Description automatically generated

* + Như ở phần versioning thì ta không thể xoá hay chỉnh sửa đối tượng input.py, lý do là ta đã bật tính năng Retention (180 days) nên tính năng Object Locking cũng được bật theo.
  + Còn nếu ta xoá nguyên 1 object thì nó vẫn hiện là đã xoá nhưng nó lưu lại mình đã xoá những gì và tất nhiên nó vẫn còn lưu trong hệ thống.

A white rectangular object with black text

Description automatically generated

A close up of a screen

Description automatically generated

* + Ta đã có 4 version.

A screenshot of a phone

Description automatically generated

* + Tiến hành remove all version.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* + Hệ thống vẫn không cho phép ta xoá bởi vì ban đầu ta đã bật tính năng Retention (180 days) dẫn đến tính năng Object Locking cũng được bật.
  + Khôi phục lại.

A screenshot of a computer

Description automatically generated

* + Xem tất cả các versions.

A screenshot of a computer

Description automatically generated

* + Bây giờ tiến hành đổi Retention từ 180 days -> 1 days thì 1 ngày sau chúng ta có thể thực hiện xoá được và giải phóng dung lượng.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Test Step 2: Để có thể xoá các object và bucket trong khi bật tính năng Object Locking thì ta sẽ thử tạo 1 bucket với chỉ các tính năng sau (không có retention):**

A screenshot of a computer

Description automatically generated

* + Upload file lên.

A close up of a logo

Description automatically generated

* + Tiến hành thử xoá file và khôi phục và xem các versions của nó.

A screenshot of a computer

Description automatically generated

* + Bây giờ, ta thực hiện xoá tất cả các version sau đó xoá bucket xem có thành công hay không.

A white background with black lines

Description automatically generated

* + Kết quả thành công và ta đã xoá thành công bucket test2.
* **Quota (Hạn mức):**
  + Use Case: Người quản trị muốn giới hạn dung lượng hoặc số lượng đối tượng có thể được lưu trữ trong một bucket.
  + Business Case: Quản lý tài nguyên hệ thống, đảm bảo không gian lưu trữ được sử dụng một cách hiệu quả và hợp lý.

Test Case:

* **Test Step 1: Thiết lập hạn mức cho bucket:**
  + Tạo một bucket với hạn mức dung lượng hoặc số lượng đối tượng.

A screenshot of a bucket

Description automatically generated

* **Test Step 2: Kiểm tra việc áp dụng hạn mức:**
  + Thử thêm đối tượng mới và xác nhận rằng khi vượt quá hạn mức, thao tác bị từ chối hoặc nhận thông báo cảnh báo.

A screenshot of a computer

Description automatically generated

A screenshot of a red and white box

Description automatically generated

* **Retention (Giữ lại):**
  + Use Case: Người dùng muốn đảm bảo rằng các đối tượng không thể bị xóa hoặc sửa đổi trong một khoảng thời gian xác định.
  + Business Case: Đáp ứng các yêu cầu pháp lý, bảo vệ dữ liệu quan trọng trong thời gian nhất định.
* Testcase: Thiết lập thời gian giữ lại cho đối tượng trong bucket và thử thay đổi hoặc xóa nó để kiểm tra xem việc giữ lại này có được áp dụng hay không.

TÌM HIỂU VỀ CÁC ĐẶC TÍNH KHÁC.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **Tuyên bố 1**:
  + **Effect**: **"Allow"** (cho phép). Tất cả các hành động với prefix là **admin:** được cho phép.
  + **Action**: **"admin:\*"**: Đây là wildcard (**\***) đại diện cho tất cả các hành động bắt đầu bằng **admin:**. Chính sách này cho phép tất cả các hành động với prefix là **admin:**.
* **Tuyên bố 2**:
  + **Effect**: **"Allow"** (cho phép). Tất cả các hành động trên AWS Key Management Service (KMS) được cho phép.
  + **Action**: **"kms:\*"**: Tất cả các hành động trên KMS đều được cho phép.
* **Tuyên bố 3**:
  + **Effect**: **"Allow"** (cho phép). Tất cả các hành động trên Amazon S3 được cho phép.
  + **Action**: **"s3:\*"**: Tất cả các hành động trên S3 đều được cho phép.
  + **Resource**: **"arn:aws:s3:::\*"**: Đây là ARN (Amazon Resource Name) đại diện cho tất cả các bucket trên Amazon S3. Chính sách này cho phép tất cả các hành động trên tất cả các bucket trên S3.

A white line with black lines

Description automatically generated with medium confidence

A close-up of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* **BandwidthMonitor**: Cho phép theo dõi thông lượng mạng, kiểm soát tải lên và tải xuống trong hệ thống lưu trữ.
* **ConsoleLog**: Cho phép truy cập và xem log (nhật ký) của bản điều khiển trên các tài nguyên S3.
* **OBDInfo**: Cung cấp thông tin chi tiết về các lỗi ghi vào hệ thống (On-Board Diagnostics).
* **Profiling**: Cho phép tiến hành phân tích, ghi chép dữ liệu chi tiết để phân tích hoạt động hệ thống.
* **Prometheus**: Cung cấp khả năng giám sát và thu thập các metric từ các target hệ thống.
* **ServerInfo**: Cho phép truy cập thông tin chi tiết về máy chủ, cấu hình, và thông tin hệ thống.
* **ServerTrace**: Cho phép theo dõi các bước đi của server, truy vết các hoạt động diễn ra trên máy chủ.
* **TopLocksInfo**: Cung cấp thông tin về các khóa đang được sử dụng (top locks) trên hệ thống lưu trữ.

A screenshot of a computer

Description automatically generated

A screenshot of a group

Description automatically generated

A screenshot of a policy report

Description automatically generated

* **s3:GetBucketLocation**: Cho phép người dùng lấy thông tin vị trí (location) của bucket trong Amazon S3. Hành động này cho phép xác định vị trí vật lý của bucket, chẳng hạn như khu vực (region) hoặc khu vực địa lý mà bucket được lưu trữ.
* **s3:GetObject**: Cho phép người dùng lấy đối tượng (object) từ bucket. Hành động này cho phép truy cập và tải về dữ liệu từ các đối tượng được lưu trữ trong bucket.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Chính sách này cấp cho người dùng tất cả các hành động (**s3:\***) liên quan đến Amazon S3 trên tất cả các tài nguyên (**arn:aws:s3:::\***). Điều này bao gồm toàn quyền truy cập và quản lý các bucket và các đối tượng bên trong các bucket. Hành động này có thể bao gồm tất cả các thao tác như tạo, xóa, sửa đổi, sao chép, tải xuống và quản lý các tài nguyên S3. Đây là một chính sách rất mạnh mẽ với quyền hạn rộng lớn đối với tất cả các tài nguyên S3 trong tài khoản.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Chính sách này cho phép thực hiện hành động **s3:PutObject**, nghĩa là người dùng được phép tạo và ghi đối tượng vào các bucket S3 được chỉ định trong tài nguyên **arn:aws:s3:::\***. Hành động này cung cấp quyền ghi dữ liệu mới vào S3, cho phép tạo và đẩy các đối tượng mới vào bất kỳ bucket nào trong tài khoản.

A screenshot of a web page

Description automatically generated

A screenshot of a computer

Description automatically generated

MinIO sử dụng Kiểm soát Truy cập Dựa trên Chính sách (PBAC) để xác định các hành động được ủy quyền và tài nguyên mà người dùng được xác thực có quyền truy cập. Mỗi chính sách mô tả một hoặc nhiều hành động và điều kiện định rõ quyền của một người dùng hoặc nhóm người dùng. PBAC của MinIO được xây dựng để tương thích với cú pháp, cấu trúc và hành vi chính sách IAM của AWS. Mỗi người dùng chỉ có thể truy cập vào những tài nguyên và hoạt động được cấp quyền một cách rõ ràng bởi vai trò tích hợp sẵn. MinIO từ chối quyền truy cập vào bất kỳ tài nguyên hoặc hành động nào khác theo mặc định.

A screenshot of a computer

Description automatically generated

* **OpenID Connect (OIDC)**: Là một giao thức xác thực dựa trên web, thường được sử dụng để quản lý việc đăng nhập và xác thực người dùng trên internet. Nó cung cấp các dịch vụ đăng nhập an toàn, hỗ trợ single sign-on (SSO) và cho phép ứng dụng và dịch vụ truy cập thông tin người dùng một cách an toàn.
* **Identity Provider (IDP)**: Là một hệ thống chịu trách nhiệm quản lý thông tin xác thực của người dùng, bao gồm việc cấp phát và xác thực token để chứng minh danh tính của họ. Các IDP như Okta, KeyCloak, Dex, Google, hoặc Facebook là những dịch vụ cung cấp xác thực người dùng thông qua các cơ chế như username/password, mã OTP, hoặc token.
* **Single-Sign On (SSO)**: Là một quy trình cho phép người dùng đăng nhập một lần vào một ứng dụng hay dịch vụ và sau đó có thể truy cập vào các ứng dụng khác mà không cần phải đăng nhập lại. Khi tích hợp với MinIO, việc cấu hình một IDP ngoài sẽ cho phép người dùng xác thực một lần trước khi truy cập MinIO và các ứng dụng khác liên quan.

A screenshot of a computer

Description automatically generated

MinIO cung cấp hỗ trợ sử dụng dịch vụ Active Directory hoặc LDAP (AD/LDAP) để quản lý danh tính người dùng từ bên ngoài. Bằng cách cấu hình một IDentity Provider (IDP) bên ngoài, MinIO cho phép quy trình Đăng nhập Một lần (SSO), trong đó các ứng dụng xác thực trước khi truy cập MinIO thông qua IDP bên ngoài.

MinIO thực hiện truy vấn tới máy chủ Active Directory/LDAP được cấu hình để xác minh thông tin đăng nhập được chỉ định bởi ứng dụng và tùy chọn trả về danh sách các nhóm mà người dùng thuộc về. MinIO hỗ trợ hai chế độ (Lookup-Bind Mode và Username-Bind Mode) để thực hiện các truy vấn này.

* **Lookup-Bind Mode**: Là phương pháp được khuyến nghị để xác minh thông tin đăng nhập AD/LDAP. Ở đây, MinIO tìm kiếm và xác minh thông tin đăng nhập của người dùng trên cơ sở dữ liệu AD/LDAP.
* **Username-Bind Mode**: Là một phương pháp lỗi thời được giữ lại chỉ vì sự tương thích ngược với các phiên bản cũ. Ở đây, MinIO sử dụng tên người dùng để trực tiếp kết nối và xác minh thông tin đăng nhập từ máy chủ AD/LDAP.

MinIO khuyến nghị sử dụng chế độ Lookup-Bind vì đây là phương pháp ưu tiên để xác thực thông tin đăng nhập từ AD/LDAP. Chế độ Username-Bind chỉ được giữ lại để hỗ trợ ngược với các phiên bản cũ của hệ thống.

A screenshot of a computer

Description automatically generated

Event Destinations trong MinIO là các địa điểm sự kiện cho phép các quản trị viên gửi thông báo đến các dịch vụ bên ngoài được hỗ trợ khi có các sự kiện cụ thể xảy ra trên đối tượng hoặc bucket. MinIO hỗ trợ các sự kiện S3 ở cấp độ bucket và đối tượng tương tự như Amazon S3 Event Notifications. Các sự kiện này bao gồm việc tạo, xóa, hoặc thay đổi đối tượng trong bucket và có thể kích hoạt các thông báo hoặc hành động khác tùy thuộc vào cấu hình của người quản trị.

A screenshot of a computer

Description automatically generated

Các loại Tier trong MinIO cho phép quản lý vòng đời của đối tượng, nơi MinIO có thể tự động chuyển đối tượng đến một "tier" lưu trữ từ xa.

MinIO hỗ trợ các loại Tier sau:

* **MinIO hoặc lưu trữ tương thích S3 khác:** Cho phép lưu trữ đối tượng trên MinIO hoặc các dịch vụ lưu trữ tương thích S3 khác.
* **Amazon S3:** Cho phép chuyển đối tượng đến lưu trữ Amazon S3.
* **Google Cloud Storage:** Cho phép chuyển đối tượng đến lưu trữ Google Cloud Storage.
* **Microsoft Azure Blob Storage:** Cho phép chuyển đối tượng đến lưu trữ Microsoft Azure Blob Storage.

Việc sử dụng các Tier này giúp tối ưu hóa lưu trữ và quản lý vòng đời của đối tượng trong hệ thống MinIO, bằng cách tự động di chuyển chúng đến các nơi lưu trữ khác nhau tùy thuộc vào quy tắc đã được thiết lập.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Đối với Replication Site, các thay đổi sau sẽ được sao chép đến tất cả các site khác:

* **Tạo và xóa bucket và đối tượng:**
* **Tạo và xóa tất cả người dùng IAM, nhóm, chính sách và ánh xạ của chúng đến người dùng hoặc nhóm:**
* **Tạo và xoá STS credentials:**
* **Tạo và xóa các tài khoản dịch vụ (trừ các tài khoản do người dùng root sở hữu):**
* **Thay đổi các tính năng của Bucket như:**
  + Chính sách Bucket
  + Thẻ Bucket
  + Cấu hình khóa đối tượng của Bucket
  + Cấu hình mã hóa của Bucket
* **Các tính năng của Bucket sau sẽ KHÔNG được sao chép:**
  + Cấu hình thông báo Bucket
  + Cấu hình vòng đời (ILM) của Bucket

IAM: Identity and Access Management (Quản lý Danh tính và Quyền Truy cập) - IAM cho phép quản lý quyền truy cập vào dịch vụ và tài nguyên của AWS.

STS: Security Token Service (Dịch vụ Mã Token Bảo mật) - STS cung cấp mã token tạm thời để xác thực và cấp quyền truy cập tới người dùng.

ILM: Lifecycle Management (Quản lý Vòng đời) - ILM cho phép quản lý tự động vòng đời của đối tượng trong bucket, bao gồm di chuyển, xóa hoặc mã hóa tự động.

### 6. Tìm hiểu thêm về MinIO Feature Overview: Object Lifecycle Management, Active-Active Replication

* **"MinIO Feature Overview: Object Lifecycle Management"**, MinIO giới thiệu về quản lý vòng đời đối tượng (Object Lifecycle Management) để giúp quản lý lượng dữ liệu ngày càng tăng trong doanh nghiệp hiện đại. MinIO cung cấp các tính năng mạnh mẽ để xử lý dữ liệu và cho phép hai lựa chọn chính khi làm việc với quản lý vòng đời đối tượng.
  + Lựa chọn đầu tiên là hết hạn (expire) đối tượng, tức là xóa nó. Ví dụ, ứng dụng kiểm tra an ninh chấp nhận các logs và metrics mới để xử lý, nhưng sau khi xử lý hoàn thành, dữ liệu gốc không cần phải được giữ trong MinIO vô thời hạn. Việc hết hạn dữ liệu giúp giữ lượng dữ liệu đang chờ xử lý ở mức thấp, tiết kiệm không gian lưu trữ.

A screen shot of a machine

Description automatically generated

* + Lựa chọn thứ hai là chuyển đổi đối tượng (transition), di chuyển nó đến một cluster khác. Điều này cho phép bạn duy trì dữ liệu một cách hiệu quả về chi phí bằng cách sử dụng hai tầng lưu trữ: nóng và ấm (hot tier and warm tier). Sau một thời gian, đối tượng sẽ chuyển đến tầng ấm nhưng vẫn có thể truy cập thông qua tầng nóng. Lý do chính của việc chia thành hai tầng này là để giữ các đối tượng ít được truy cập thường xuyên trên tầng ấm, thường có dung lượng lưu trữ chậm hơn, giải phóng không gian trên tầng nóng cho các đối tượng hoạt động hơn sử dụng bộ nhớ nhanh hơn. Minh họa bằng việc giữ bản chụp cơ sở dữ liệu, bản chụp gần đây sẽ được sử dụng thường xuyên hơn để làm việc với dữ liệu mới. Tuy nhiên, chúng ta có thể cần giữ hàng năm dữ liệu cho mục đích kiểm toán, mặc dù kiểm toán này ít được thực hiện hơn. Vì vậy, sau một thời gian, việc chuyển dữ liệu cũ sang tầng lưu trữ chậm hơn nhưng vẫn giữ dữ liệu sẵn có là hợp lý. Cuối cùng, khi dữ liệu không còn hữu ích, chúng ta có thể xóa nó, giữ cho không gian lưu trữ tầng ấm được sạch sẽ. Đó là lý do tại sao MinIO hỗ trợ quản lý vòng đời đối tượng.

A red arm holding a black text

Description automatically generated

A graph of different types of water

Description automatically generated with medium confidence

* **“MinIO Feature Overview: Active-Active Replication”,** MinIO giới thiệu tính năng Active-Active Replication, tập trung vào việc đảm bảo sẵn có liên tục dữ liệu - một yếu tố quan trọng đối với doanh nghiệp hiện đại. Việc sao chép dữ liệu đến nhiều địa điểm trên toàn cầu đồng thời và thời gian thực là cần thiết để đảm bảo tính liên tục và khả năng sẵn có cao nhất.

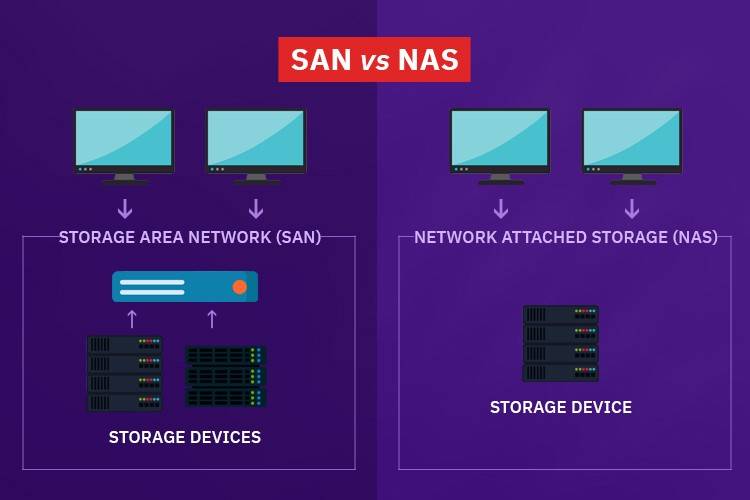
A line art of a city

Description automatically generated

A map of the world

Description automatically generated

* Active-Active Replication của MinIO cho phép các ứng dụng ghi đồng thời dữ liệu vào hai hoặc nhiều địa điểm với đồng bộ thời gian thực, khả năng này không có trong các hệ thống SAN và NAS truyền thống. Dữ liệu trong một cụm được sao chép đầy đủ giữa các địa điểm và ứng dụng có thể ghi dữ liệu vào bất kỳ địa điểm nào.



* Khi một địa điểm bị mất, các địa điểm còn lại tiếp tục hỗ trợ ứng dụng mà không có thời gian chết hoặc cần phải xây dựng lại. Không chỉ các đối tượng được sao chép, bất kỳ biến đổi nào đối với siêu dữ liệu đối tượng cũng được duy trì đồng bộ giữa các địa điểm. MinIO hỗ trợ cả sao chép đồng bộ và không đồng bộ, với khả năng đồng bộ trước khi hoàn tất hoạt động gốc và khả năng hoàn tất hoạt động gốc trước khi đặt đối tượng vào hàng đợi sao chép.

A blue background with white text and icons

Description automatically generated

A diagram of a computer

Description automatically generated with medium confidence

* Nó cũng xem xét các tình huống thất bại thực tế, như khi một địa điểm ngừng hoạt động tạm thời, khi mất kết nối giữa các địa điểm, hoặc khi đối tượng được sửa đổi đồng thời ở cả hai địa điểm. MinIO cung cấp giải pháp cho mỗi tình huống này bằng cách duy trì hàng đợi và hỗ trợ quá trình tự lành lỗi khi các địa điểm trở lại hoạt động.

A screenshot of a screen

Description automatically generatedA diagram of different colored balls

Description automatically generated with medium confidence

A screenshot of a computer screen

Description automatically generated A screenshot of a graph

Description automatically generated

A screenshot of a graph

Description automatically generated A screenshot of a computer game

Description automatically generated

A screenshot of a computer game

Description automatically generated A screenshot of a computer screen

Description automatically generated

* Cuối cùng, nó cung cấp các giải pháp cho các tình huống xấu nhất như mất hoàn toàn một trung tâm dữ liệu do thảm họa. MinIO cho phép khôi phục toàn bộ cụm dữ liệu khi một địa điểm mới sẵn sàng, cũng như khi thêm các địa điểm mới. Đây là cách MinIO đảm bảo khả năng sẵn có dữ liệu quan trọng của doanh nghiệp hiện đại.

A screenshot of a graph

Description automatically generated A screen shot of a computer

Description automatically generatedA screenshot of a screen

Description automatically generated