International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

CLOUD STORAGE OBJECT ACCESSING USING SIGNED URL

Ms.T. NANDHINI

Supervisor
Assistant Professor, Department of IT,
KSR Institute for Engineering andTechnology,
Tirchengode

1. Mr.J.KARTHICK 2.Mr.S.SRIDHARAN
3. Mr.P.TAMILSELVAN 4.Mr.T.VEERASAKTHIVEL
Student, IV Year B.Tech IT
KSR Institute for Engineering and Technology,Tirchengode

Abstract:- In today's world, cloud storage has become an integral part of our lives, and businesses rely heavily on cloud storage providers to store their data. However, accessing these cloud storage objects securely can be a challenging task. One solution to this problem is to use signed URLs, which provide temporary access to cloud storage objects with an added layer of security. The project aims to create a web application that allows users to securely access cloud storage objects through signed URLs. The web application will be designed to work with multiple cloud storage providers such as Amazon Web Services (AWS), Google Cloud Storage, and Microsoft Azure. The web application will have two main components: a user interface and a backend service. The user interface will provide users with a simple way to access their cloud storage objects. Users will be able to upload, download, and delete objects from their cloud storage accounts. The web application will be designed with security in mind. All user data will be encrypted both at rest and in transit. The application will also implement access control measures to ensure that users can only access objects that they are authorized to access.

INTRODUCTION

The Cloud Storage Object Accessing using Signed URL project aims to provide a secure and user-friendly way for individuals and businesses to access their cloud storage objects using signed URLs. The project will be designed to work with multiple cloud storage providers, including Amazon Web Services (AWS), Google Cloud Storage, and Microsoft Azure. The project will have two main components: a user interface and a backend service. The user interface will provide users with a simple way to access their cloud storage objects, while the backend service will handle all the interactions with the cloud storage providers**.** When a user requests access to an object, the backend service will generate a signed URL that includes the user's access credentials and a time-limited token. The signed URL will provide the user with temporary access to the requested object, and the token will ensure that the access is valid only for a limited period of time. The web application will be designed with security in mind. All user data will be encrypted both at rest and in transit. The application will also implement access control measures to ensure that users can only access objects that they are authorized to access.In addition to the core functionality, the web application will also include features such as search and metadata management. Users will be able to search for objects using keywords and metadata tags. They will also be able to add and edit metadata tags associated with their objects.

I. LITERATURE SURVEY

Cloud computing is a trending paradigm that combines several computing concepts and technologies of the Internet to create a platform for more agile, cost effective and reliable model for the public users, business applications and IT infrastructure. There are various requirements that need to be addressed by Cloud Service Provider (CSP) for enabling the cloud services to the users such as security, performance, availability, inerrability, customization with minimal cost. If any of these requirements are not met, then the user wishes to switch from current CSP to a new CSP. To achieve that the user has to download all the digital assets, services, IT resources and applications from one CSP and upload into another CSP. This process has many issues like security, vendor management, technical integration, requirement of time and energy resources, etc; The first one being a major concern which we are addressing it in this paper. Here we propose a secure data migration technique to migrate the data from one cloud storage system to another cloud storage system. The proposed approach comprises of mutual authentication, blended with key splitting and sharing methods that ensure pre-migration authentication. The migration of data is then performed by encrypting with symmetric keys, which are shared using RSA Cryptosystem. The security factors such as confidentiality, authorization, authenticity, integrity is ensured by this technique. The proposed methodology is implemented and validated on two OpenStack servers using the Object Storage Service accessed by swift client.

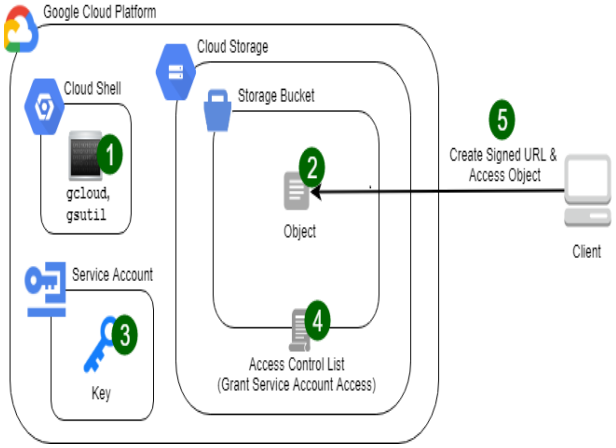
II. PROPOSED SYSTEM

Authentication and Authorization: The system would require a mechanism for authenticating and authorizing users or applications to generate signed URLs. This could involve integrating with an existing authentication and authorization system, such as OAuth or OpenID Connect.

Signed URL Generation-The system would generate signed URLs using a private key and the appropriate parameters, such as

the expiration time and the resource being accessed. The generated signed URL would be returned to the user or application requesting access to the object.

Access Control: The system would provide fine-grained access control capabilities by allowing access to specific objects or parts of objects using the signed URL. This would be based on the permissions granted to the user or application generating the signed URL.Auditability:The system would log access requests made using the signed URL, providing an audit trail for compliance or security purposes. Integration with Existing Applications: The system would be designed to be easily integrated with existing applications, allowing those applications to securely access objects in cloud storage using signed URLs.Cost Optimization-The system would optimize costs by minimizing the number of requests made to the cloud storage platform, as signed URLs can be used to provide temporary access to resources without requiring additional authentication requests.Security:The system would provide strong security for accessing objects in cloud storage by ensuring that only authorized users or applications canutes to build.



Figno:1 overall architecture

A. The following modules are present in this paper

2.1 Create a Cloud Storage Bucket.

In the GCP Console, select the project where you want to create the bucket. Click on the "Navigation menu" icon (☰) in the top-left corner of the screen. Expand the "Cloud Storage" section and click on "BUCKET". Click the "Create bucket" button at the top of the page. Enter a unique name for your bucket. Bucket names must be globally unique across all projects and comply with the DNS naming conventions (e.g., signed access).Choose the location where you want to store your data. This determines the geographic location where your data will be physically stored. You can choose from a variety of locations around the world. Choose a default storage class for your bucket. The default storage class determines the pricing and availability characteristics of your bucket. You can choose from a variety of storage classes, including Multi-Regional Storage, Regional Storage, Near line Storage, and Cold line Storage. Click the "Create" button to create your bucket.

2.2 GENERATE SERVICE ACCOUNT KEY

To generate a service account key in Google Cloud Platform (GCP), follow these steps: Open the GCP Console. Select the project for which you want to create a service account key from the project dropdown menu at the top of the page.Navigate to the IAM & Admin page by clicking on the menu icon in the top left corner of the page and selecting "IAM & Admin" from the -list of available services. Click on "Service Accounts" in the left-hand menu. Click on the name of the service account for which you want to create a key. Click on the "Add Key" button at the top of the page and select "Create New Key" from the dropdown menu. Choose the key format you want to use. You can choose between JSON and P12 formats. JSON is the recommended format for most use cases.

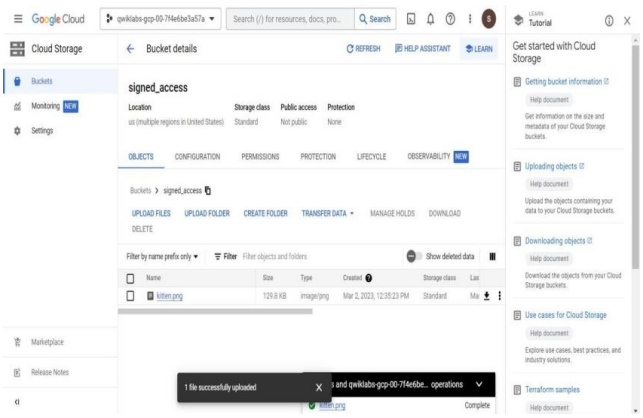
2.3 UPLOAD KEY IN PROJECT

Upload service account key in the project. Open the GCP Console. Click on the Cloud Shell icon in the top right corner of the page. It looks like a >_ symbol. Click the three dots right side corner. Click the upload option Select your key in your computer. Click upload to upload your service account key in your project**.**

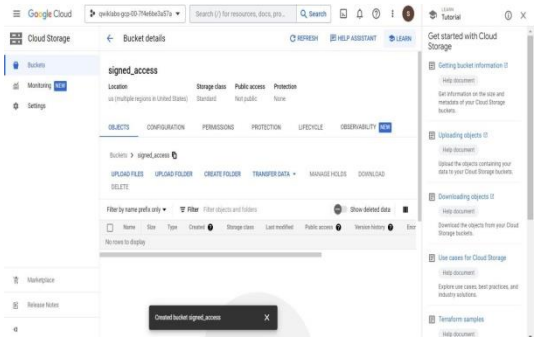
2.4 CREATE SIGNED URL

Open the GCP Console. Click on the Cloud Shell icon in the top right corner of the page. It looks like a >_ symbol. In the Cloud Shell window, make sure you have the correct project set by running the following command: gcloudconfig set project [PROJECT_ID]Replace [PROJECT_ID] with your GCP project ID. Run the following command to generate a signed URL: gsutilsignurl -d [EXPIRATION_TIME][PATH_TO_KEY_FILE] gs://[BUCKET_NAME]/[OBJECT_NAME]Replace [EXPIRATION_TIME] with the desired expiration time for the URL, in the format "HH:MM:SS". Replace [PATH_TO_KEY_FILE] with the file path to the JSON key file for your service account. Replace [BUCKET_NAME] with the name of the Cloud Storage bucket

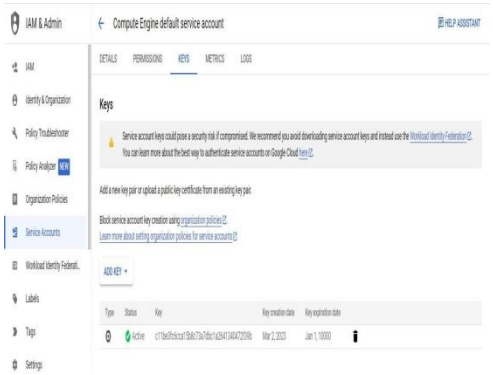
III RESULTS



FIGNO:2 CREATE BUCKETS



Figno:3 **UPLOAD FILES IN CLOUD STORAGE BUCKET**



Figno:4 CREATE PRIVATE KEY FOR DEFAULT SERVICE ACCOUNT

IV-CONCLUSION

In conclusion, the use of cloud storage object accessing using signed URLs has become increasingly important for ensuring secure data access in cloud-based applications and services. It enables temporary access to cloud storage objects without compromising the overall security of the data. cloud storage object accessing using signed URLs is a powerful security feature that offers a variety of benefits to cloud-based applications and services. it enables secure data sharing, media streaming, IoT device data transfers, cloud- based application access, and data archiving

FUTURE ENHANCEMENT

In future Enhancement With the continuous growth of cloud computing, the future of cloud storage object accessing using signed URLs looks promising. Its potential applications are vast, ranging from secure data sharing to IoT devices and media streaming.If you're interested in working on a project related to cloud storage object accessing using signed URLs, there are many opportunities to explore this topic further and innovate in this area of cloud security.

REFERENCES

1.Google Cloud Storage Signed URLs documentation: https://cloud.google.com/storage/docs/access-control/signed-urls

2.AWS S3 Pre-Signed URLs documentation: https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSigne dURL.html

3.Microsoft Azure Shared Access Signature documentation: https://docs.microsoft.com/en-us/azure/storage/common/storage-sas-overview

4.Digital Ocean Spaces Signed URLs documentation: https:/www.digitalocean.com/docs/spaces/resources/s3-sdk-examples/generate- presigned-url/