iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IX **Month of publication:** September 2021

DOI: https://doi.org/10.22214/ijraset.2021.37946

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

Automated Deployment of Data Lake

Ganavi J[1], Amith S Bharadwaj[2]
[1]*Altiostar, Bangalore, Karnataka*
[2]*Sakha Global, Bangalore, Karnataka*

Abstract: ***A Data Lake is a central location that can store all your structured and unstructured data, no matter the source or format. Automated deployment for data lake solution is an automated reference implementation that deploys a highly available, cost-effective data lake architecture on the AWS Cloud along with a user-friendly console for searching and requesting datasets. The solution automatically configures the core AWS services necessary to easily tag, search, share, transform, analyse, and govern specific subsets of data across a company or with other external users. The solution deploys a console that users can access to search and browse available datasets for their business needs.***
Keywords: Data Lake, Cloud Computing, Aws, Ec2, S3, Athena, Glue, Cloud formation.

I. INTRODUCTION

A Data Lake is a central location that can store all your structured and unstructured data, no matter the source or format. When utilized with specific data processing tools (like Hadoop), Data Lake scalable solutions help you efficiently manage, analyze and extract all relevant and available data. You gain the ability to quickly validate data against your objectives and make more rapid and accurate business decisions. [1]

Automated deployment for data lake solution is an automated reference implementation that deploys a highly available, cost-effective data lake architecture on the AWS Cloud along with a user-friendly console for searching and requesting datasets. The solution automatically configures the core AWS services necessary to easily tag, search, share, transform, analyze, and govern specific subsets of data across a company or with other external users. The solution deploys a console that users can access to search and browse available datasets for their business needs. [1]

II. PROBLEM STATEMENT

A. Existing System

In the existing system there is a centralized repository called a Data Lake in which the data is stored for analysis and retrieval and the data retrieval process is that that automated and core services of the data lake and other configurations needs to be handled manually. [1]

B. Limitations of Existing System

In the existing system whenever the data lake is deployed by the user from organizations the core services of data lake needs to be configured manually and this would consume a lot of time and efforts. [1]

C. Proposed System

A data lake is a new and increasingly popular way to store and analyse data because it allows companies to manage multiple data types from a wide variety of sources, and store this data, structured and unstructured, in a centralized repository. The AWS Cloud provides many of the building blocks required to help customers implement a secure, flexible, and cost- effective data lake. These include AWS managed services that help ingest, store, find, process, and analyse both structured and unstructured data. To support our customers as they build data lakes, AWS offers the data lake solution, which is an automated reference implementation that deploys a highly available, cost-effective data lake architecture on the AWS Cloud. [1]

D. Advantages of Proposed System

The solution leverages the security, durability and scalability of Amazon S3 to manage the persistent catalogue of organizational datasets and Amazon DynamoDB to manage corresponding metadata. Once a dataset is catalogued, its attributes and descriptive tags are available to search on. User can search and browse available dataset in the solution console, and create a list of data they require access to. The solution keeps track of datasets a user selects and generate a manifest file with secure access links to desired content when the user checks out.

E. Features of Data Lake

1) *Data Lake reference implementation:* Leverage this data lake solution out-of-the-box, or as a reference implementation that you can customize to meet unique data management, search, and processing needs. [1]
2) *User Interface:* The solution automatically creates an intuitive, web-based console UI hosted on Amazon S3 and delivered by Amazon Cloud Front. Access the console to easily manage data lake users, data lake policies, add or remove data packages, search data packages, and create. [1]
3) *Command line interface:* Use the provided CLI or API to easily automate data lake activities or integrate this solution into existing data automation for dataset ingress, egress, and analysis. [1]
4) *Managed storage layer:* Secure and manage the storage and retrieval of data in a managed Amazon S3 bucket, and use a solution-specific AWS Key Management Service (KMS) key to encrypt data at rest. [1]
5) *Data access flexibility:* Leverage pre-signed Amazon S3 URLs, or use an appropriate AWS Identity and Access Management (IAM) role for controlled yet direct access to datasets in Amazon S3. [1]
6) *Data transformation and analysis:* Upload datasets with searchable metadata that integrates with AWS Glue and Amazon Athena to transform and analyse the data. [1]
7) *Federation sign in:* Optionally, you can enable users to sign in through a SAML identity provider (IdP) such as Microsoft Active Directory Federation Services (AD FS). [4]

III.SYSTEM DESIGN

A. High Level Design

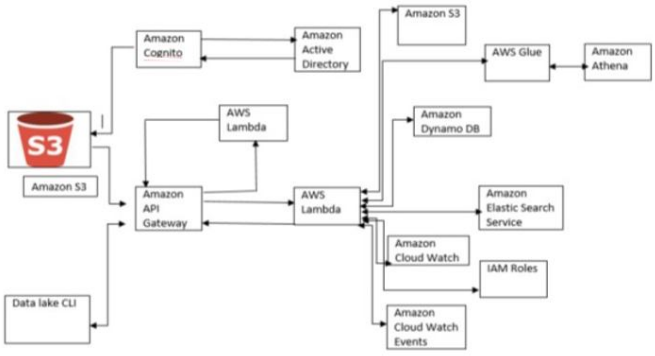


Fig. 1 An architecture of the features within a data lake

Fig. 1 Show an architecture of the features within a data lake and several examples of how data enters a data lake.

This architecture, which is entirely serverless and backed by Amazon S3, lets you scale your data lake to easily accommodate any data size platform. Additionally, the components presented in this diagram can be secured via IAM controls and service policies. Amazon S3 is the canonical source for objects in your RWE data lake. This enables you to secure and protect the sensitive information that often resides within an RWE platform. [4]

You track and search metadata that is associated with these objects in a data catalog built on Amazon Elasticsearch Service and Amazon DynamoDB.

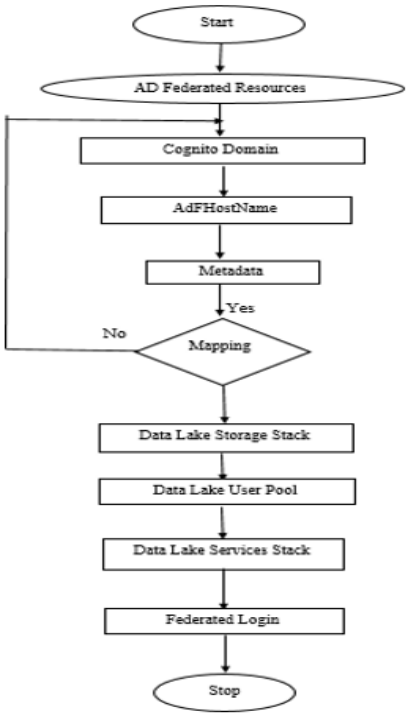
1) Streaming (e.g., wearables), structured, and unstructured data is acquired from myriad devices and sources. Depending on the data size, you might use AWS IoT (streaming), AWS Storage Gateway (mid- size/continual batch), and AWS Snowball (large legacy datasets, such as imaging). AWS IoT writes to Amazon Kinesis Firehose, which transforms the telemetry data in-flight to land both transformed and raw data in Amazon S3. [1]
2) When data lands in Amazon S3 buckets, an AWS Lambda function is invoked (either by trigger or manually). [1]
3) This Lambda function writes to a data catalogue that is fronted by Amazon API Gateway. The data catalog contains metadata about all the object data in Amazon S3, as well as data that resides in databases. [1]
4) An AWS Lambda function on the other side of API Gateway writes the appropriate metadata about the objects, such as the study that the data was generated from, into Amazon Elasticsearch service. [1]

The solution creates a data lake console and deploys it into an Amazon S3 bucket configured for static website hosting, and configures an Amazon CloudFront distribution to be used as the solution's console entrypoint. During initial configuration, the solution also creates a default administrator role and sends an access invite to a customer-specified email address. Note that if you deploy the federated stack, you must manually create user and admin groups. [4]

IV. FLOWCHART

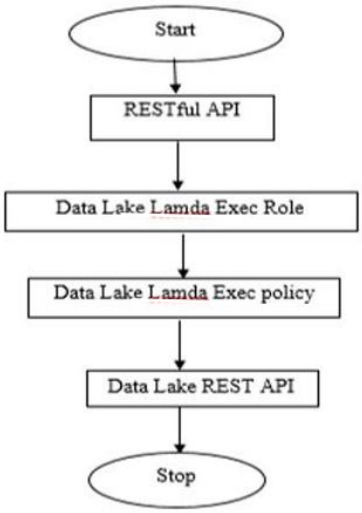
A. Data Lake deploy federated template

This template is used to launch a version of the solution that is ready to integrate with your existing SAML identity provider such Microsoft Active Directory. [4]



B. Data Lake API template

This template deploys the Amazon API Gateway resources.



ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IX Sep 2021- Available at www.ijraset.com

V. HARDWARE REQUIREMENTS

A. *Processor*: Intel Core i3
B. *Processor Speed*: 1.70 GHz
C. *RAM*: 4GB
D. *Storage Space*: 2GB

VI. SOFTWARE REQUIREMENTS

A. *Operating System*: Windows v8/8.1/10.
B. *Technology*: Amazon Web Services
C. *Software*: Npm 4.x,node.js 8.10.
D. *Client side requirements*: Retrieval, Analysis, Data catalog, Data governance.
E. *Server side requirements*: Security, Authentication, Authorization, Storage.

VII. IMPLEMENTATION

A. *Pseudocode*

1) *DataLake Deploy Template:* The below code is a template for datalake deployment, this template is designed to take input parameters such as data lake administrator name,

AWS Template Descr Data Lake resources.
Parameters:
AdministratorName:
Type: String
Description: Name of Data Lake administrator
AllowedPattern : ".+
AdministratorEmail:
Type: String
Description: Email address for Data Lake Admin
AllowedPattern: "^[_ z]{2,})$"
CognitoDomain:
Type: String
Description: Prefixed domain names can only contain lower-case letters, numbers, and hyphens.
AllowedPattern : -z 9-]

2) *DataLake Storage Template:* The below template is designed to create storage space in simple storage service(S3) an create cognito domain name to manage user pools. *[5]*

```
AWSTemplateFormatVersion: "2010-09-09"

Mappings:
    RegionMap:
        us-east-1:
            "InstanceType":"m4.large.elastic search"
            "DedicatedMasterType":"t2.sma ll.elasticsearch"
        us-east-2:
            "InstanceType": "m4.large.elasticsearch"
            "DedicatedMasterType":"t2.sma ll.elasticsearch"
    Resources:
        DataLakeSettingsDynamo:
            Type: "AWS::DynamoDB::Table"
            Deletion Policy: "Delete"
        Properties:
```

```
        AttributeDefinitions:
            AttributeName: "setting_id"
            AttributeType: "S"
Outputs:
    EsCluster:
        Description: "Elasticsearch cluster domain endpoint"
        Value:!GetAttDataLakeElasticse archCluster.DomainEndpoint
    CognitoKibanaIdentityPool:
        Description:"Cognito kibana identity pool"
        Value:!RefCognitoKibanaIdentityPool
```

F. Testing Analysis

The following table shows the test case which checks if the Data Lake responds to the inputs provided to it.

Table I
Response Of Data Lake To Inputs

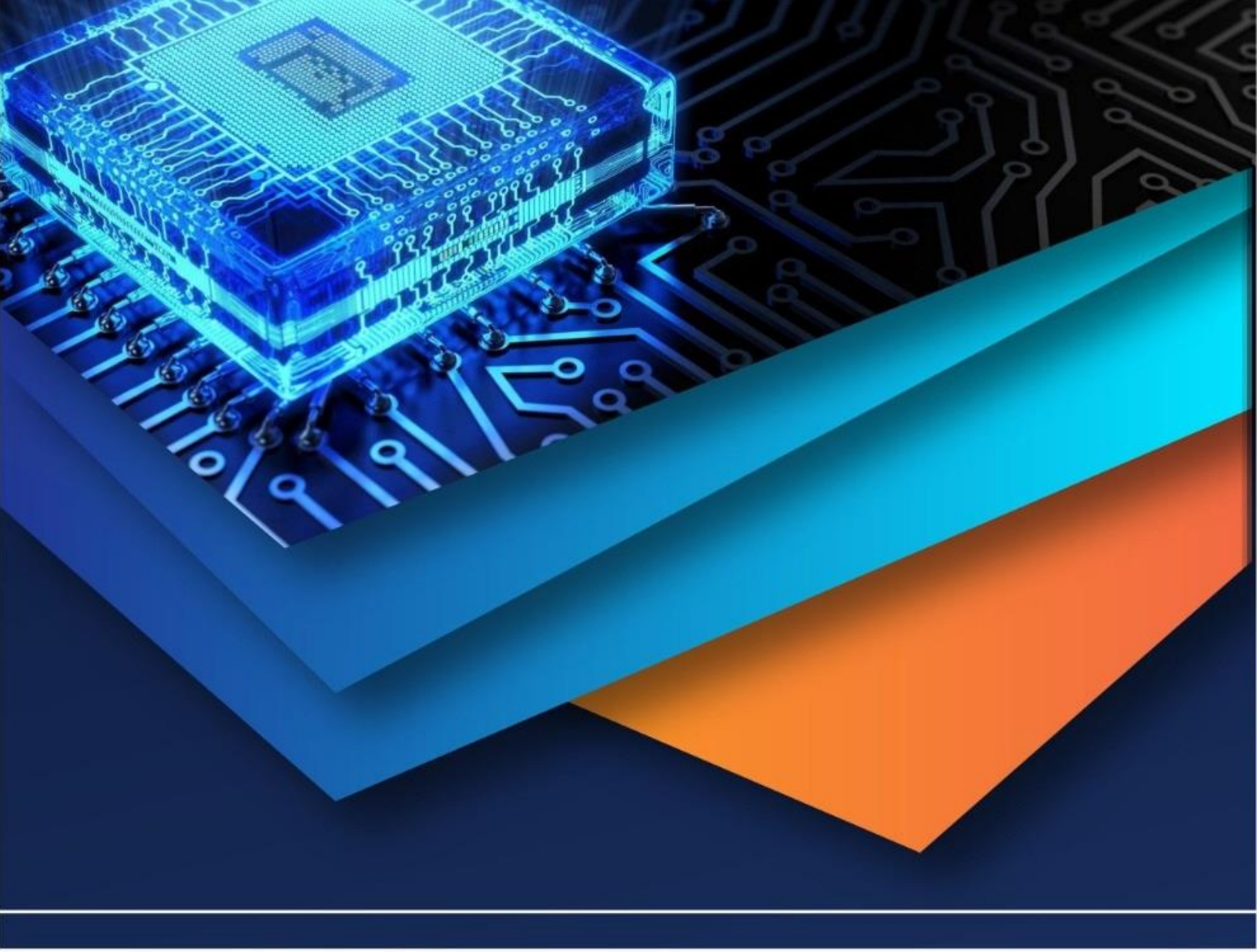
Sl No. of test case :	1
Name of test :	Deployment code
Item / Feature being tested	Deploy model
Sample Input	Create stack on Cloud Formation
Expected Output:	Creates stack and deploys templates on amazon simple storage service (S3).
Actual Output:	Creates stack and deploys templates on amazon simple storage service(S3)
Remarks:	Test succeeded

VIII. CONCLUSIONS

The solution uses AWS Cloud Formation to deploy the infrastructure components supporting this data lake reference implementation. At its core, this solution implements a data lake API, which leverages Amazon API Gateway to provide access to data lake micro services, (AWS Lambda functions). These micro services provide the business logic to create data packages, upload data, search for existing packages, add interesting data to a cart, generate data manifests, and perform administrative functions.
These micro services interact with Amazon S3, AWS Glue, Amazon Athena, Amazon Dynamo DB, Amazon ES, and Amazon Cloud Watch Logs to provide data storage, management, and audit functions. [4]

REFERENCES

[1] https://aws.amazon.com/.
[2] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html
[3] https://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html
[4] http://docs.aws.amazon.com/solutions/latest/data-lake-solution/welcome.html
[5] https://github.com/awslabs/aws-data-lake-solution/blob/master/deployment/data-lake-storage.template
[6] *Putting the Data Lake to work*, CITO Research, April 2014
[7] Tomcy John and Pankaj misra, *Data Lake for Enterprises – Leveraging Lamdba Architecture for Building Enterprises Data Lake*, Published, May 2017
[8] Mark Harring, *Connected Data Ponds: The evolution of Data Lakes*, Hortonworks, September 08, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY