Data Lake Solution

AWS Implementation Guide

November 2016

Contents

About This Guide

This implementation guide discusses architectural considerations and configuration steps for deploying the data lake solution on the Amazon Web Services (AWS) Cloud. It includes links to AWS CloudFormation templates that launch, configure, and run the AWS compute, network, storage, and other services required to deploy this solution on AWS, using AWS best practices for security and availability.

The guide is intended for IT infrastructure architects, administrators, and DevOps professionals who have practical experience architecting on the AWS Cloud.

Overview

Many Amazon Web Services (AWS) customers require a data storage and analytics solution that offers more agility and flexibility than traditional data management systems. A *data lake* is a new and increasingly popular way to store and analyze data because it allows companies to store all of their data, structured and unstructured, in a centralized repository. An effective data lake should provide low-cost, scalable, and secure storage, and support search and analysis capabilities on a variety of data types.

The AWS Cloud provides many of the building blocks required to help customers implement a secure, flexible, and cost-effective data lake. 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 along with a user-friendly console for searching and requesting datasets. The solution is intended to address common customer pain points around conceptualizing data lake architectures, and automatically configures the core AWS services necessary to easily tag, search, share, and govern specific subsets of data across a company or with other external users. This solution allows users to catalog new datasets, and to create data profiles for existing datasets in Amazon Simple Storage Service (Amazon S3) with minimal effort.

The data lake solution stores and registers datasets of any size in their native form in the secure, durable, highly-scalable Amazon S3. Additionally, user-defined tags are stored in Amazon DynamoDB to add business-relevant context to each dataset. The solution enables companies to create simple governance policies to require specific tags when datasets are registered with the data lake. Users can browse available datasets or search on dataset attributes and tags to quickly find and access data relevant to their business needs. The solution keeps track of the datasets a user selects in a cart (similar to an online shopping cart) and then generates a manifest file with secure access links to the desired content when the user checks out.

Cost

You are responsible for the cost of the AWS services used while running this reference deployment. As of the date of publication, the cost for running the data lake solution with default settings in the US East (N. Virginia) Region is less than $1 per hour. This reflects Amazon API Gateway, AWS Lambda, Amazon DynamoDB, and Amazon Elasticsearch Service (Amazon ES) charges.

This cost does not include variable data storage and outbound data-transfer charges from Amazon S3 and Amazon CloudWatch Logs for data that the solution manages. For full details, see the pricing webpage for each AWS service you will be using in this solution.

Architecture Overview

Deploying this solution with the **default parameters** builds the following environment in the AWS Cloud.

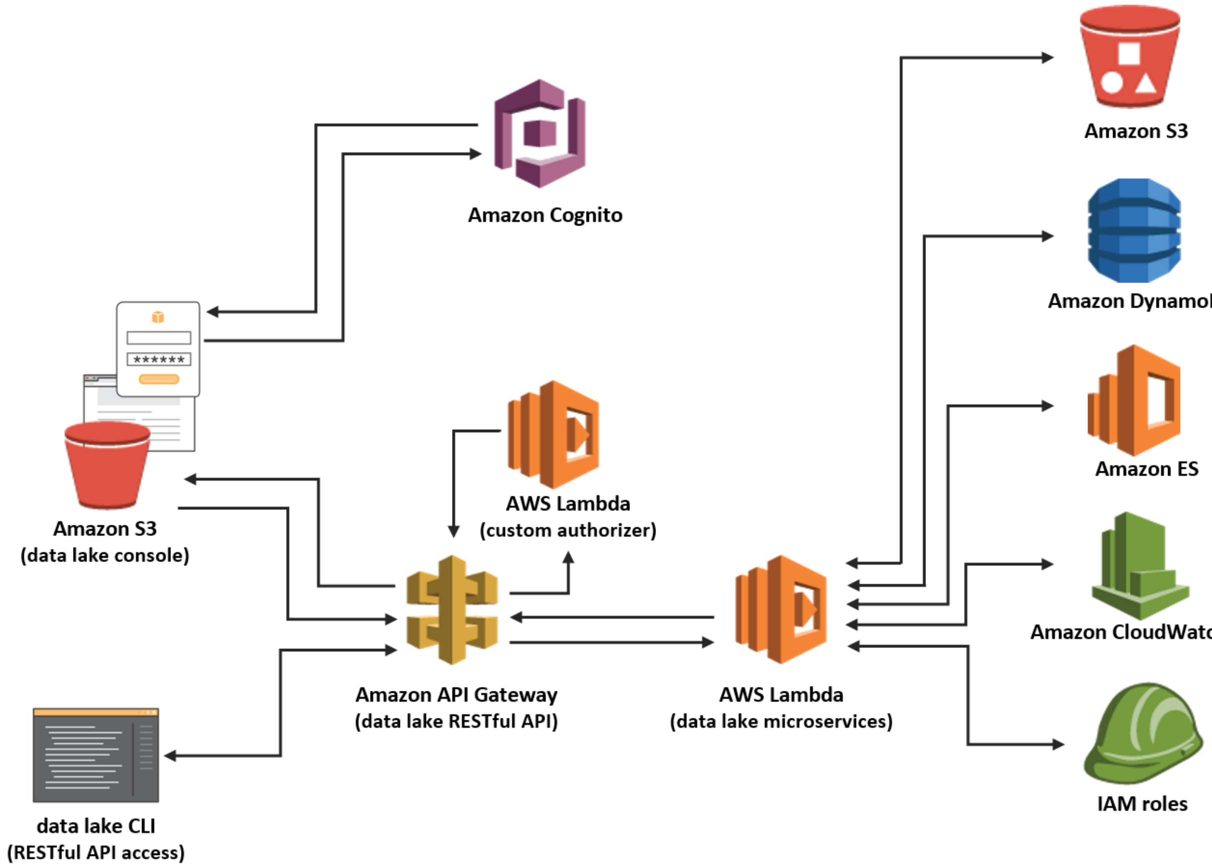

Figure 1: Data lake solution architecture on AWS

The solution uses AWS CloudFormation 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 microservices, (AWS Lambda functions). These microservices 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 microservices interact with Amazon S3, Amazon DynamoDB, Amazon ES, and Amazon CloudWatch Logs to provide data storage, management, and audit functions.

The solution creates a console and deploys it into an Amazon S3 bucket configured for static website hosting. During initial configuration, the solution also creates a default Administrator role and sends an access invite to a customer-specified user email. The solution uses an Amazon Cognito user pool to manage user access to the console and the data lake API. See Appendix A for detailed information on each of the solution's architectural components.

Solution Features

This data lake solution provides the following features:

- **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.
- **User interface:** The solution automatically creates an intuitive, web-based console UI hosted on Amazon S3. Access the console to easily manage data lake users, data lake policies, add or remove data packages, search data packages, and create manifests of datasets for additional analysis.
- **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.
- **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 (AWS KMS) key to encrypt data at rest.
- **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.

AWS CloudFormation Templates

This solution uses AWS CloudFormation to automate the deployment of the data lake solution on the AWS Cloud. It includes the following AWS CloudFormation templates, which you can download before deployment:

View template

data-lake-deploy.template: This is the main template used to launch the data lake solution and all associated components. You can also customize the template based on your specific needs. This template, in turn, launches the following nested stacks:

- **data-lake-storage.template:** This template deploys the Amazon S3, Amazon Elasticsearch Service, and Amazon DynamoDB components of the solution.
- **data-lake-services.template:** This template deploys the AWS Lambda microservices and the necessary IAM roles and policies. In addition, it deploys the AWS KMS resources for the solution.
- **data-lake-api.template:** This template deploys the Amazon API Gateway resources.

Automated Deployment

Before you launch the automated deployment, please review the architecture, configuration, network security, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the data lake solution into your account.

Time to deploy: Approximately 30 minutes

What We'll Cover

The procedure for deploying this architecture on AWS consists of the following steps. For detailed instructions, follow the links for each step.

Step 1. Launch the stack

- Launch the AWS CloudFormation template into your AWS account.
- Enter values for required parameters: **Stack Name, Administrator Name**, **Administrator Email**, and **Access IP Address.**
- Review the other template parameters, and adjust if necessary.

Step 2. Log in to the Data Lake Console

- Log in with the URL and temporary password sent to the Administrator email.
- Review the solution's online guide.

Step 1. Launch the Stack

The AWS CloudFormation template automatically deploys the data lake solution on the AWS Cloud.

> **Note**: You are responsible for the cost of the AWS services used while running this solution. See the Cost section for more details. For full details, see the pricing webpage for each AWS service you will be using in this solution.

1. Log in to the AWS Management Console and click the button to the right to launch the *data-lake-deploy* AWS CloudFormation template.

   Launch Solution

   You can also download the template as a starting point for your own implementation.
2. The template is launched in the US East (N. Virginia) Region by default. To launch the data lake solution in a different AWS Region, use the region selector in the console navigation bar.

> **Note**: This solution uses AWS Lambda and Amazon Cognito which are currently available in specific AWS Regions only, therefore you must launch this solution in an AWS Region where these services are available. [1]

3. On the **Select Template** page, verify that you selected the correct template and choose **Next**.
4. On the **Specify Details** page, assign a name to your data lake solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Administrator Name	<Requires input>	The user name for the initial solution Administrator. After the solution is deployed, this Administrator can create and manage other users, including additional Administrators.
Administrator Email	<Requires input>	A valid email associated with the Administrator user.
Access IP Address	<Requires input>	The source IP address of the Administrator(s) who can access Amazon ES cluster to perform any necessary management functions.
Send Anonymous Usage Data	`Yes`	Send anonymous data to AWS to help us understand solution usage and related cost savings across our customer base as a whole. To opt out of this feature, choose `No`. For more information, see Appendix B.

6. Choose **Next**.
7. On the **Options** page, you can specify tags (key-value pairs) for resources in your stack and set additional options, and then choose **Next**.
8. On the **Review** page, review and confirm the settings. Be sure to check the box acknowledging that the template will create AWS Identity and Access Management (IAM) resources with custom names.
9. Choose **Create** to deploy the stack.

   You can view the status of the stack in the AWS CloudFormation console in the **Status** column. After the stack launches, the three nested stacks will be launched in the same AWS Region. Once all of the stacks and stack resources have successfully launched, you will see the message CREATE_COMPLETE. This can take 25 minutes or longer.

[1] For the most current service availability by AWS Region, see https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/

Step 2. Log in to the Data Lake Console

After the data lake stack launch completes the Administrator will receive an email that contains the URL to the data lake console and a temporary password.

> **Note:** This email will be sent from *no-reply@verificationemail.com*. Check your email configuration to make sure you do not block or filter emails from this domain.

1. Click the link in the email to open the solution console, and then log in with your email address and the temporary password.
2. You will be prompted to set a new password, and then you will be signed in to the console.
3. In the top navigation bar, choose **Support** to open the online guide.

   Explore the guide subsections (**User Guide**, **Admin Guide**, and **CLI**) for specific instructions and examples.

Security

The AWS Cloud provides a scalable, highly reliable platform that helps customers deploy applications and data quickly and securely. When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS, which can reduce your operational burden. For more information about security on AWS, visit the AWS Security Center.

User Authorization

Authorized users access the data lake using the solution-generated console, the data lake CLI, or direct calls to the data lake APIs. Users log in to the data lake console with their user name (by default, their email) and password. Authentication to the console is managed in an Amazon Cognito user pool.

Requests to the data lake API are HTTPS based and must be signed with an access key (access key and secret access key combination) to confirm the user's identity. Administrators can grant API access on an individual user basis. If a user is granted API access, an access key is generated to identify that user's calls to the data lake API. Each user has the ability to generate their own secret access keys to allow them to work with the data lake CLI or make direct API calls.

See Appendix A for additional component-level security information.

Additional Resources

AWS service documentation

- AWS CloudFormation
- AWS Lambda
- Amazon S3
- Amazon DynamoDB
- Amazon API Gateway
- Amazon Cognito
- Amazon Elasticsearch Service
- AWS Key Management Service
- Amazon CloudWatch

AWS webpages

- What is a Data Lake?
- Big Data on AWS
- AWS Answers: Data Lakes on AWS

Appendix A: Architectural Components

AWS KMS Key

The data lake AWS KMS key (alias: **datalake**) is created to provide encryption of all dataset objects that the solution owns and stores in Amazon S3. Additionally, the AWS KMS key is used to encrypt the secret access key in each user's Amazon Cognito user pool record for API access to the data lake.

Amazon S3

The solution uses a default Amazon S3 bucket to store datasets and manifest files associated with packages that users upload to the data lake. Additionally, the bucket stores the manifest files generated for a user when they check out their cart, which is a collection of packages. All access to this bucket (get and put actions from the package and manifest microservices) is controlled via signed URLs. All objects stored in this bucket are encrypted using the data lake AWS KMS key.

A second Amazon S3 bucket hosts the data lake console. This console is a static website that uses Amazon Cognito for user authentication.

Amazon Cognito User Pool

The data lake console is secured for user access with Amazon Cognito and provides an administrative interface for managing data lake users through integration with Amazon Cognito user pools. Only Administrators can create user accounts and send invitations to those users. When an Administrator creates a new user, he/she will assign the user one of the following roles, with the associated permissions:

- **Member:** The member role can perform non-administrative actions within the data lake. These actions include the following:
  - View and search all packages in the data lake
  - Add, remove, and generate manifests for packages in their cart
  - Create, update, and delete packages they created
  - Create and update metadata on the packages they created
  - Add and remove datasets from the packages they created
  - View their data lake profile and API access information
  - Generate a secret access key if an Administrator has granted them API access
- **Admin:** The admin role has full access to the data lake. The admin role can perform the following actions in addition to the member role actions:
  - Create user invitations
  - Update, disable, enable, and delete data lake users
  - Create, revoke, enable, and disable a user's API access
  - Update data lake settings
  - Create, update, and delete governance settings

Data Lake API and Microservices

The data lake API receives requests via HTTPS. When an API request is made, Amazon API Gateway leverages a custom authorizer (AWS Lambda function) to ensure that all requests are authorized.

The data lake microservices is a series of AWS Lambda functions that provide the business logic and data access layer for all data lake operations. Each AWS Lambda function assumes an IAM role with least privilege access (minimum permissions necessary) to perform its designated functions. The following sections outline each data lake microservice.

Admin Microservice

The *data-lake-admin-service* is an AWS Lambda function that processes data lake API requests sent to the `/admin/*` endpoints. The admin microservice handles all administrative services including user management, general settings, governance settings, API keys, and role authorization for all operations within the data lake.

Cart Microservice

The *data-lake-cart-service* is an AWS Lambda function that processes data lake API requests sent to the `/cart/*` endpoints. The cart microservice handles all cart operations including item lists, adding items, removing items, and generating manifests for user carts.

Manifest Microservice

The *data-lake-manifest-service* is an AWS Lambda function that manages import and export of manifest files. The manifest microservice uploads import manifest files, which allows existing Amazon S3 content to be bulk imported into a package. It also generates export manifest files for each package in a user's cart at checkout.

Package Microservice

The *data-lake-package-service* is an AWS Lambda function that processes data lake API requests sent to `/packages/*` endpoints. The package microservice handles all package operations including list, add package, remove package, update package, list metadata, add metadata, update metadata, list datasets, add dataset, remove dataset, and process manifest.

Search Microservice

The *data-lake-search-service* is an AWS Lambda function that process data lake API requests sent to `/search/*` endpoints. The search microservice handles all search operations including query, index document, and remove indexed document.

Profile Microservice

The *data-lake-profile-service* is an AWS Lambda function that processes data lake API requests sent to `/profile/*` endpoints. The profile microservice handles all profile operations for data lake users, including get and generate secret access key.

Logging Microservice

The *data-lake-logging-service* is an AWS Lambda function that interfaces between the data lake microservices and Amazon CloudWatch Logs. Each microservice sends operations and access events to the logging service, which records the events in Amazon CloudWatch Logs. You can access this log (*datalake/audit-log*) in the CloudWatch console.

Amazon DynamoDB Tables

The data lake solution uses Amazon DynamoDB tables to persist metadata for the data packages, settings, and user cart items. The following tables are provisioned during deployment and only accessed via the data lake microservices:

- **data-lake-packages**: persistent store for data package title and description
- **data-lake-metadata:** persistent store for metadata tag values associated with packages
- **data-lake-datasets:** persistent store for dataset pointers to Amazon S3 objects
- **data-lake-cart:** persistent store for user cart items
- **data-lake-keys:** persistent store for user access key ID references
- **data-lake-settings:** persistent store for data lake configuration and governance settings

Amazon Elasticsearch Service Cluster

The solution uses an Amazon Elasticsearch Service cluster to index data lake package data for searching. The cluster is accessible only by the search microservice and an IP address that the customer designates during initial deployment.

Appendix B: Collection of Anonymous Data

This solution includes an option to send anonymous usage data to AWS. We use this data to better understand how customers use this solution to improve the services and products that we offer. When enabled, the following information is collected and sent to AWS:

- **Solution ID:** The AWS solution identifier
- **Unique ID (UUID):** Randomly generated, unique identifier for each data lake solution deployment
- **Timestamp:** Data-collection timestamp
- **Cluster Size:** Size of the Amazon Elasticsearch cluster the solution will deploy

Note that AWS will own the data gathered via this survey. Data collection will be subject to the AWS Privacy Policy. To opt out of this feature, set the **Send Anonymous Usage Data** parameter to `No`.

Send Us Feedback

We welcome your questions and comments. Please post your feedback on the AWS Solutions Discussion Forum.

You can visit our GitHub repository to download the templates and scripts for this solution, and to share your customizations with others.

Document Revisions

Date	Change	In sections
November 2016	Initial release	-

Notices

This document is provided for informational purposes only. It represents AWS's current product offerings and practices as of the date of issue of this document, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.